ECHELON®

FT 3120® / FT 3150®
Smart Transceiver Data Book

005-0139-01B

Smart Transceivers, Neuron Chips, and other OEM Products were not designed for use in equipment or systems which involve danger to human health or safety or a risk of property damage and Echelon assumes no responsibility or liability for use of the Smart Transceivers or Neuron Chips in such applications.

Parts manufactured by vendors other than Echelon and referenced in this document have been described for illustrative purposes only, and may not have been tested by Echelon. It is the responsibility of the customer to determine the suitability of these parts for each application.

Printed in the United States of America.

Echelon Corporation
www.echelon.com

Table Of Contents

Chapter 3 - Input/Output Interfaces **3-1**

List of Tables

List of Figures

1

Introduction

1.1 Introduction

This manual provides detailed technical specifications on the electrical interfaces, mechanical interfaces, and operating environment characteristics for the FT 3120® and FT 3150® Smart Transceivers. This manual also provides guidelines for migrating applications to an FT 3120 or FT 3150 Smart Transceiver-based device using a LonBuilder® or NodeBuilder®development tool.

In some cases, vendor sources are included in this manual to simplify the task of integrating the FT 3120 and FT 3150 Smart Transceivers with application electronics.

There is a list of related documentation at the end of this chapter in section 1.5, *Related Documentation*. The documents listed in this section can be found on Echelon's website at www.echelon.com unless otherwise noted.

1.2 Audience

The *FT 3120/FT 3150 Smart Transceiver Databook* provides specifications and user instructions for FT 3120 and FT 3150 Smart Transceiver customers, and users of network interfaces based on the FT 3120 and FT 3150 Smart Transceivers.

1.3 Product Overview

The FT 3120 and FT 3150 Smart Transceivers integrate a Neuron® 3120 or Neuron 3150 network processor core, respectively, with a free topology twisted pair transceiver to create a low cost, smart transceiver on a chip. Combined with Echelon's high performance FT-X1 Communication Transformer, the FT 3120 and FT 3150 Smart Transceivers set new benchmarks for performance, robustness, and low cost. Ideal for use in LONWORKS® devices designed for building, industrial, transportation, home, and utility automation applications, the FT 3120 and FT 3150 Smart Transceivers can be used in both new product designs and as a means of cost reducing existing devices.

The integral transceiver is fully compatible with the TP/FT-10 channel and can communicate with devices using Echelon's FTT-10A Free Topology Transceiver, and with the addition of suitable DC isolation capacitors, the LPT-10 Link Power

Transceiver. The free topology transceiver supports polarity insensitive cabling using a star, bus, daisy-chain, loop, or combined topologies. This frees the installer from the need to adhere to a strict set of wiring rules. Free topology wiring reduces the time and expense of device installation by allowing the wiring to be installed in the most expeditious and cost-effective manner. It also simplifies network expansion by eliminating restrictions on wire routing, splicing, and device placement.

The FT 3120 Smart Transceiver is a complete system-on-a-chip that is targeted at cost-sensitive and small form factor designs that require up to 4Kbytes of application code. The Neuron 3120 core operates at up to 40MHz, and includes 4Kbytes of EEPROM and 2Kbytes of RAM. The Neuron firmware is pre-programmed in an on-chip ROM. The application code is stored in the embedded EEPROM memory and may be updated over the network. The FT 3120 Smart Transceiver is offered in a 32-lead SOIC package as well as a compact 44-lead TQFP package.

The FT 3150 Smart Transceiver includes a 20MHz Neuron 3150 core, 0.5Kbytes of EEPROM and 2Kbytes of RAM. Through its external memory bus, the FT 3150 Smart Transceiver can address up to 58Kbytes of external memory, of which 16Kbytes of external non-volatile memory is dedicated to the Neuron system firmware. The FT 3150 Smart Transceiver is supplied in a 64-lead TQFP package.

The embedded EEPROM may be written up to 10,000 times with no data loss. Data stored in the EEPROM will be retained for at least 10 years.

Three different versions of the FT 3120 and FT 3150 Smart Transceivers are available to meet a wide range of applications and packaging requirements. See the table below for product offerings and descriptions.

Table 1.1 FT 3120 and FT 3150 Smart Transceiver Product Offerings

Smart Transceiver IC Product Number	Model Number	Maximum input clock	EEPROM (Kbytes)	RAM (Kbytes)	ROM (Kbytes)	External memory interface	IC Package
FT 3120-E4S40	14210-500	40MHz	4Kbytes	2Kbytes	12Kbytes	No	32 SOIC
FT 3120-E4P40	14220-800	40MHz	4Kbytes	2Kbytes	12Kbytes	No	44 TQFP
FT 3150-P20	14230-450	20MHz	0.5Kbytes	2Kbytes	N/A	Yes	64 TQFP

The FT 3120 and FT 3150 Smart Transceivers provide 11 I/O pins which may be configured to operate in one or more of 34 predefined standard input/output modes. Combining a wide range of I/O models with two on-board timer/counters enables the FT 3120 and FT 3150 Smart Transceivers to interface to application circuits with minimal external logic or software development.

The FT 3120 and FT 3150 Smart Transceivers can be easily interfaced to other host MCUs via Echelon’s ShortStack™ or MIP firmware. When used with the ShortStack or MIP firmware, the Smart Transceiver enables any OEM product with a host microcontroller to quickly and inexpensively become a networked, Internet-accessible device. The ShortStack firmware uses an SCI or SPI serial interface to communicate between the host and the Smart Transceiver. The MIP firmware uses a high performance parallel or dual-ported RAM interface.

The FT 3120 and FT 3150 Smart Transceivers are supplied with the FT-X1 transformer, a patent pending external communication transformer. This transformer enables operation in the presence of high frequency common mode noise on unshielded twisted pair networks. Properly designed devices can meet the rigorous Level 3 requirements of EN 61000-4-6 without the need for a network isolation choke.

The transformer also offers outstanding immunity from magnetic field noise, eliminating the need for protective magnetic field shields in most applications. The transformer is provided in a potted, 6-pin, through-hole plastic package.

A typical FT 3120 or FT 3150 Smart Transceiver-based device requires a power source, crystal, and I/O circuitry. See Figure 1.1 for a typical FT 3120 / FT 3150 Smart Transceiver-based device.

The FT 3120 and FT 3150 Smart Transceivers are compatible with Echelon's LPT-10 Link Power Transceiver, and they can communicate with each other on a single twisted pair cable. This capability provides an inexpensive means of interfacing to devices whose current or voltage requirements would otherwise exceed the capacity of the link power segment. When equipped with an FT 3120 or FT 3150 Smart Transceiver and DC blocking capacitors, these devices can be operated from a local power supply without the need for additional electrical isolation from the link power network.

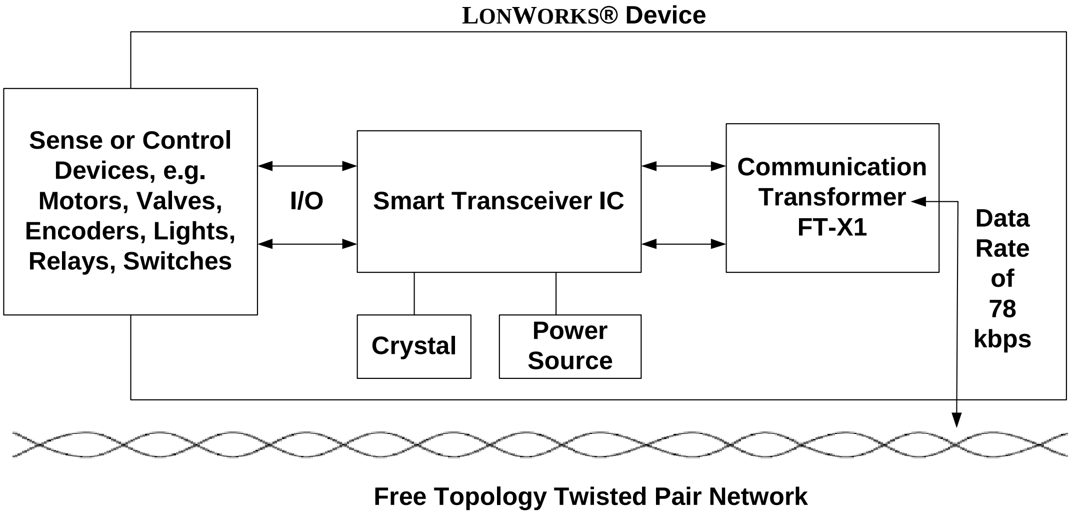

LONWORKS® Device
Sense or Control Devices, e.g. Motors, Valves, Encoders, Lights, Relays, Switches
I/O
Smart Transceiver IC
Communication Transformer FT-X1
Data Rate of 78 kbps
Crystal
Power Source
Free Topology Twisted Pair Network

Figure 1.1 Typical FT 3120 / FT 3150 Smart Transceiver-based Device

The FT 3120 and FT 3150 Smart Transceivers also provide electrical isolation for I/O devices that are grounded, allowing such devices to be used on a link power network segment. In many applications, some I/O devices are grounded, either to meet functional requirements or safety regulations. The FT-X1 transformer electrically isolates the device from the segment, allowing I/O circuitry to be grounded without impairing communications.

A twisted pair channel may be composed of multiple segments separated by EIA 709.1 routers or physical layer repeaters. A physical layer repeater may be designed using FTT-10A transceivers (the FT 3120 and FT 3150 Smart Transceivers cannot be used as physical layer repeaters). The FTT-10A transceiver includes a physical layer repeater feature that allows LONWORKS data to be exchanged between network segments by interconnecting two or more FTT-10A transceivers. This allows a twisted pair network to grow inexpensively to encompass many more devices or longer wire distances than would otherwise be possible. Refer to the *LONWORKS FTT-10A Free Topology Transceiver User's Guide* for more information on this.

The FT 3120 and FT 3150 Smart Transceivers are designed to comply with both FCC and EN 55022 EMI requirements, minimizing time-consuming and expensive testing.

1.4 Free Topology Technology Overview

A conventional control system using bus topology wiring (such as RS-485) consists of a network of sensors and actuators that are interconnected using a shielded twisted wire pair. In accordance with RS-485 guidelines, all of the devices must be wired in a bus topology to limit electrical reflections and ensure reliable communications. There is a high cost associated with installing and maintaining the cable plant that links together the devices of an RS-485-based control system. Bus topology wiring is more time consuming and expensive to install, because the installer is unable to branch or star the wiring where convenient. All devices must be connected directly to the main bus.

The best solution to reduce installation and maintenance costs and to simplify system modifications is to use a free topology communications system. Echelon's free topology transceiver technology offers such a solution, providing an elegant and inexpensive method of interconnecting the different elements of a distributed control system.

A free topology architecture allows the installer to wire the control devices with virtually no topology restrictions. Power is supplied by a local +5VDC power supply located at each device as shown in Figure 1.2.

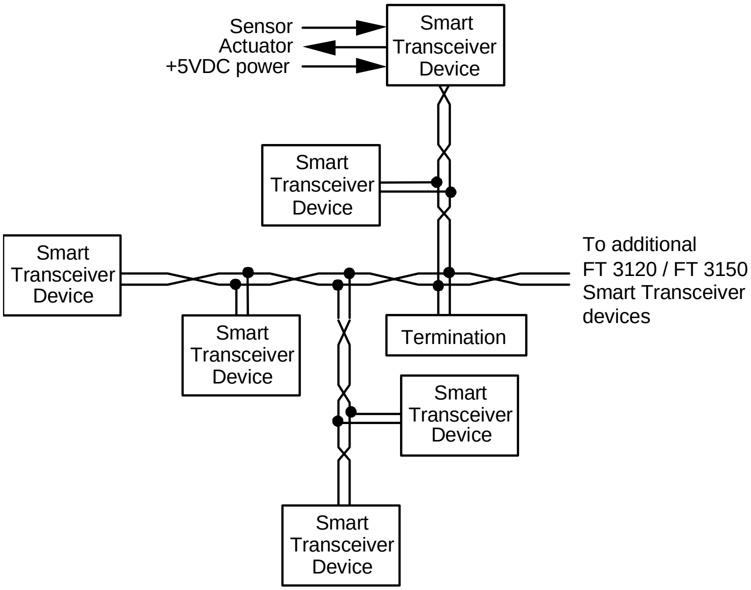
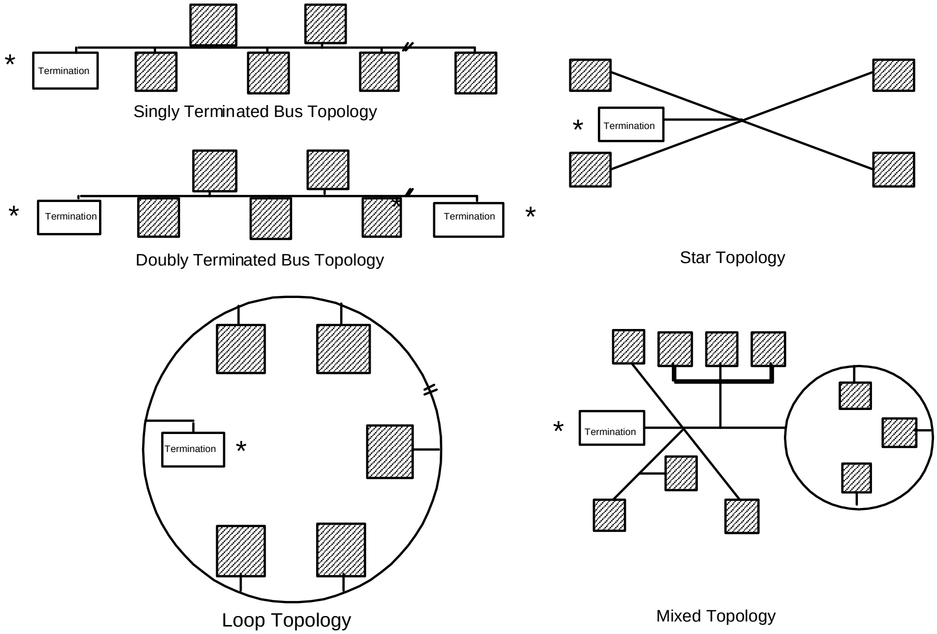


Figure 1.2 Free Topology Transceiver System

Unlike bus wiring designs, the free topology FT 3120 and FT 3150 Smart Transceivers use a wiring scheme that supports star, loop, and/or bus wiring (see Figure 1.3). This design has many advantages:

1. The installer is free to select the method of wiring that best suits the installation, reducing the need for advanced planning and allowing last minute changes at the installation site.
2. If installers have been trained to use one style of wiring for all installations, free topology technology can be introduced without requiring retraining.
3. Retrofit installations with existing wiring plants can be accommodated with minimal, if any, rewiring. This capability ensures that FT 3120 and FT 3150 Smart Transceiver technology can be adapted to both old and new projects.
4. Free topology permits FT 3120 and FT 3150 Smart Transceiver systems to be expanded in the future by simply tapping into the existing wiring where it is most convenient to do so. This reduces the time and expense of system expansion, and from the customer's perspective, keeps down the life cycle cost of the free topology network.



Singly Terminated Bus Topology

Doubly Terminated Bus Topology

Star Topology

Loop Topology

Mixed Topology

(* The actual termination circuit will vary by topology.)

Figure 1.3 Typical Wiring Topologies Supported by the FT 3120 and FT 3150 Smart Transceivers System Content

1.5 Related Documentation

The following Echelon documents are suggested reading:

LONWORKS SMX Transceiver Datasheet (003-0336-01)

LonBuilder User's Guide (078-0001-01)

NodeBuilder User's Guide (078-0141-01)

Neuron C Programmer's Guide (078-0002-01)

LonBuilder Hardware Guide (078-0003-01)

LONMARK™ Layers 1-6 Interoperability Guidelines (078-0014-01)

LONMARK™ Application Layer Interoperability Guidelines (078-0120-01)

FT 3120 and FT 3150 Smart Transceiver Datasheet (003-0337-01)

LONWORKS FTT-10A Free Topology Transceiver data sheet (003-0312-01)

LONWORKS Custom Node Development engineering bulletin (005-0024-01)

LPI-10 Link Power Interface Module User's Guide (078-0104-01)

LPT-10 Link Power Transceiver User's Guide (078-0105-01)

Junction Box and Wiring Guidelines for Twisted Pair LONWORKS Networks engineering bulletin (005-0023-01)

EIA-709.1 Control Network Protocol Specification (distibuted by Global Engineering Documents: http://global.ihs.com).

2

Hardware Resources

The first members of the Echelon Smart Transceiver family are the FT 3120 and FT 3150 Smart Transceivers. The FT 3150 Smart Transceiver supports external memory for more complex applications, while the FT 3120 Smart Transceiver is a complete system on a chip. The major hardware blocks of both processors are the same, except where noted; see Table 2.1 and Figure 2.1.

Table 2.1 Comparison of Smart Transceivers

Characteristic	FT 3150 Smart Transceiver	FT 3120 Smart Transceiver
RAM Bytes	2,048	2,048
ROM Bytes	—	12,288
EEPROM Bytes	512	4,096
16-Bit Timer/Counters	2	2
External Memory Interface	Yes	No
Package	64 pin TQFP	32 pin SOIC 44 pin TQFP

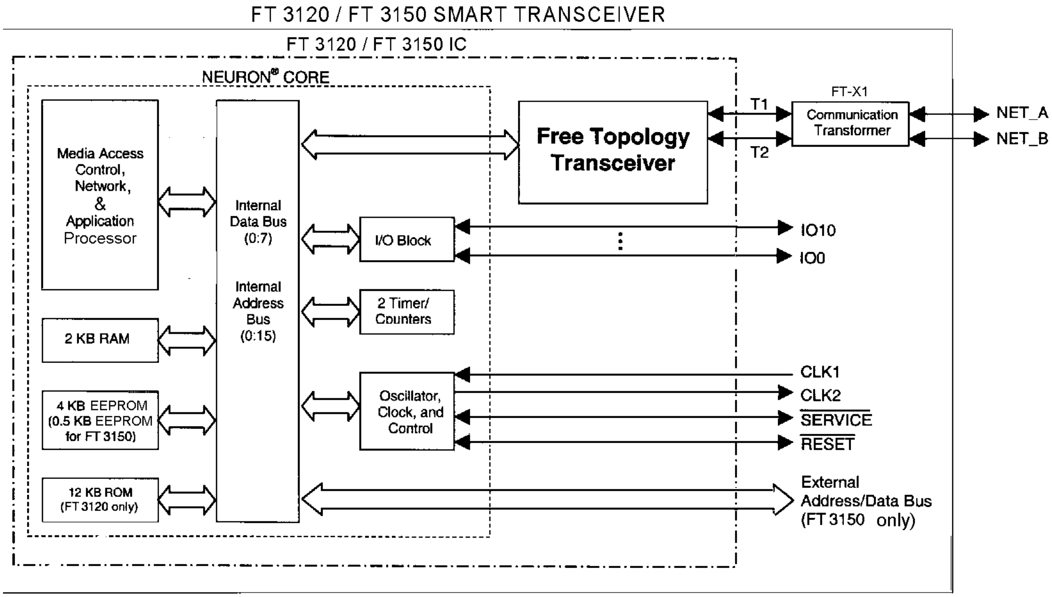

FT 3120 / FT 3150 SMART TRANSCEIVER
FT 3120 / FT 3150 IC
NEURON® CORE
Media Access Control, Network, & Application Processor
Internal Data Bus (0:7)
Internal Address Bus (0:15)
Free Topology Transceiver
T1
T2
FT-X1
Communication Transformer
NET_A
NET_B
I/O Block
IO10
IO0
2 Timer/ Counters
2 KB RAM
4 KB EEPROM (0.5 KB EEPROM for FT 3150)
Oscillator, Clock, and Control
CLK1
CLK2
SERVICE
RESET
12 KB ROM (FT 3120 only)
External Address/Data Bus (FT 3150 only)

Figure 2.1 Smart Transceiver Block Diagram

2.1 Neuron Processor Architecture

The Neuron core is composed of three processors. These processors are assigned to the following functions by the Neuron firmware.

Processor 1 is the MAC layer processor that handles layers 1 and 2 of the 7-layer LonTalk® protocol stack. This includes driving the communications subsystem hardware and executing the media access control algorithm. Processor 1 communicates with Processor 2 using network buffers located in shared RAM memory.

Processor 2 is the network processor that implements layers 3 through 6 of the LonTalk protocol stack. It handles network variable processing, addressing, transaction processing, authentication, background diagnostics, software timers, network management, and routing functions. Processor 2 uses network buffers in shared memory to communicate with Processor 1, and application buffers to communicate with Processor 3. These buffers are also located in shared RAM memory. Access to them is mediated with hardware semaphores to resolve contention when updating shared data.

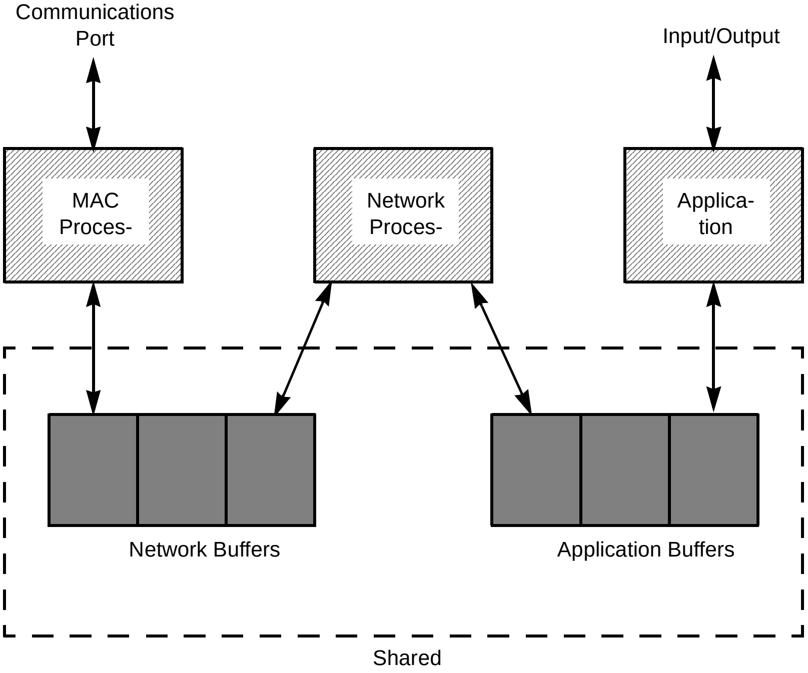


Figure 2.2 Processor Organization Memory Allocation

Processor 3 is the application processor. It executes the code written by the user, together with the operating system services called by user code. The primary programming language used by applications is Neuron C, a derivative of the ANSI C language optimized and enhanced for LONWORKS distributed control applications. The major enhancements are the following (see the *Neuron C Programmer's Guide* for details):

- A network communication model, based on *functional blocks* and *network variables*, that simplifies and promotes data sharing between like and disparate devices.
- A network configuration model, *based on functional blocks and configuration properties*, that facilitates interoperable network configuration tools.
- A type model based on standard and user *resource files* that expands the market for interoperable devices by simplifying the integration of devices from multiple manufacturers.
- An extensive set of I/O drivers that support the I/O capabilities of the Neuron core.
- Powerful event driven programming extensions that provide easy handling of network, I/O, and timer events.

The support for all these capabilities is part of the Neuron firmware, and does not need to be written by the programmer.

Each of the three identical processors has its own register set (Table 2.2), but all three processors share data, ALUs (arithmetic logic units) and memory access circuitry (Figure 2.3). On the FT 3150 Smart Transceiver, the internal address, data, and R/$\overline{\text{W}}$ signals are reflected on the corresponding external lines when utilized by any of the internal processors. Each CPU *minor* cycle consists of *three system clock* cycles, or phases; each system clock cycle is two input clock cycles. The minor cycles of the three processors are offset from one another by one system clock cycle, so that each processor can access memory and ALUs once during each instruction cycle. Figure 2.3 shows the active elements for each processor during one of the three phases of a minor cycle. Therefore, the system pipelines the three processors, reducing hardware requirements without affecting performance. This allows the execution of three processes in parallel without time-consuming interrupts and context switching.

Table 2.2 Register Set

Mnemonic	Bits	Contents
FLAGS	8	CPU Number, Fast I/O Select, and Carry Bit
IP	16	Next Instruction Pointer
BP	16	Address of 256-Byte Base Page
DSP	8	Data Stack Pointer Within Base Page
RSP	8	Return Stack Pointer Within Base Page
TOS	8	Top of Data Stack, ALU Input

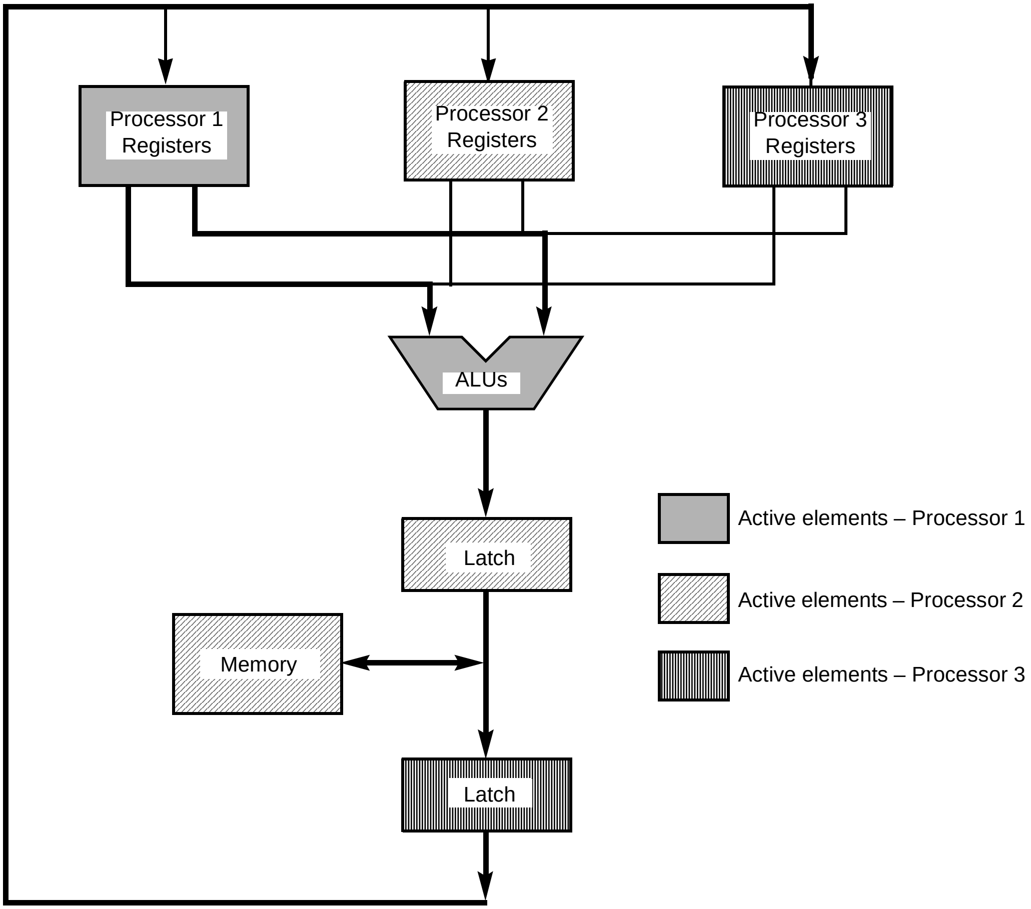

Processor 1 Registers
Processor 2 Registers
Processor 3 Registers
ALUs
Latch
Memory
Latch
Active elements – Processor 1
Active elements – Processor 2
Active elements – Processor 3

Figure 2.3 Processor/Memory Activity During One of the Three System Clock Cycles of a Minor Cycle

The architecture is stack-oriented; one 8-bit wide stack is used for data references, and the ALU operates on the TOS (Top of Stack) register and the next entry in the data stack which is in RAM. A second stack stores the return addresses for CALL instructions, and may also be used for temporary data storage. This stack architecture leads to very compact code. Tables 2.3, 2.4, and 2.5 outline the instruction set.

Figure 2.4 shows the layout of a base page, which may be up to 256 bytes long. Each of the three processors uses a different base page, whose address is given by the contents of the BP register of that processor. The top of the data stack is in the 8-bit TOS register, and the next element in the data stack is at the location within the base page at the offset given by the contents of the DSP register. The data stack grows from low memory towards high memory. The assembler shorthand symbol NEXT refers to the contents of the location (BP+DSP) in memory, which is not an actual processor register.

Pushing a byte of data onto the data stack involves the following steps: incrementing the DSP register, storing the current contents of TOS at the address (BP+DSP) in memory, and moving the byte of data to TOS.

Popping a byte of data from the data stack involves the following steps: moving TOS to the destination, moving the contents of the address (BP+DSP) in memory to TOS, and decrementing the DSP register.

The return stack grows from high memory towards low memory. Executing a subroutine call involves the following steps: storing the high byte of the instruction pointer register IP at the address (BP+RSP) in memory, decrementing RSP, storing the low byte of IP at the address (BP+RSP) in memory, decrementing RSP, and moving the destination address to the IP register.

Similarly, returning from a subroutine involves the following steps: incrementing RSP, moving the contents of (BP+RSP) to the low byte of the IP register, incrementing RSP, and moving the contents of (BP+RSP) to the high byte of IP.

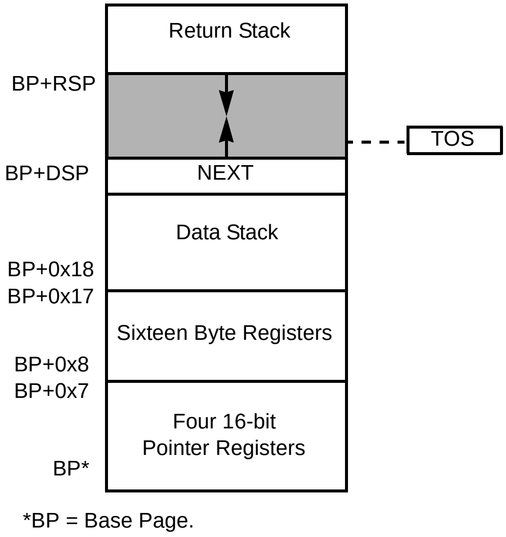


Figure 2.4 Base Page Memory Layout

A processor instruction cycle is three system clock cycles, or six input clock (CLK1) cycles. Most instructions take between one and seven processor instruction cycles. At an input clock rate of 40MHz, instruction times vary between 0.15 µs and 1.05 µs. Execution time scales inversely with the input clock rate. The formula for instruction time is:

```
<Instruction Time> = <# Cycles> x 6 / <Input Clock>
```

Tables 2.3, 2.4, and 2.5 list the processor instructions, their timings (in cycles) and sizes (in bytes). This is provided for purposes of calculating the execution time and size of code sequences. All programming of the Smart Transceiver is done with Neuron C using a LonBuilder or NodeBuilder development tool. The Neuron C compiler can optionally produce an assembly listing, and examining this listing can help the programmer to optimize his/her Neuron C source code.

Table 2.3 Program Control Instructions

Mnemonic	Cycles	Size (bytes)	Description	Comments
NOP	1	1	No operation	
SBR	1	1	Short unconditional branch	Offset 0 to 15
BR/BRC/ BRNC	2	2	Branch, branch on (not) carry	Offset -128 to +127
SBRZ/SBRNZ	3	1	Short branch on TOS (not) zero	Offset 0 to 15. Drops TOS
BRF	4	3	Unconditional branch far	Absolute address
BRZ/BRNZ	4	2	Branch on TOS (not) zero	Offset -128 to +127. Drops TOS
RET	4	1	Return from subroutine	Drops two bytes from return stack
BRNEQ	4/6	3	Branch if TOS not equal (taken/not taken)	Offset -128 to +127. Drops TOS if equal
DBRNZ	5	2	Decrement [RSP] and branch if not zero	Offset -128 to +127. If not taken, drops one byte from return stack
CALLR	5	2	Call subroutine relative	Offset -128 to +127. Pushes two bytes to return stack
CALL	6	2	Call subroutine	Address in low 8KB. Pushes two bytes to return stack
CALLF	7	3	Call subroutine far	Absolute address. Pushes two bytes to return stack

Table 2.4 Memory/Stack Instructions

Mnemonic	Cycles	Size (bytes)	Comments / Effective Address (EA)
PUSH TOS	3	1	Increment DSP, duplicate TOS into NEXT
DROP TOS	3	1	Move NEXT to TOS, decrement DSP
DROP_R TOS	6	1	Move NEXT to TOS, decrement DSP, return from call
PUSH (NEXT, DSP, RSP, FLAGS)	4	1	Push processor register
POP (DSP, RSP, FLAGS)	4	1	Pop processor register
DROP NEXT	2	1	Decrement DSP
DROP_R NEXT	5	1	Decrement DSP and return from call
PUSH/POP !D	4	1	Byte register [8 to 23]
PUSH !TOS	4	1	EA = BP + TOS, push byte to NEXT
POP !TOS	4	1	EA = BP + TOS, pop byte from NEXT
PUSH [RSP]	4	1	Push from return stack to data stack, RSP unchanged
DROP [RSP]	2	1	Increment RSP
PUSHS #literal	4	1	Push short literal value [0 to 7]
PUSH #literal	4	2	Push 8-bit literal value [0 to 255]
PUSHPOP	5	1	Pop from return stack, push to data stack
POPPUSH	5	1	Pop from data stack, push to return stack
LDBP address	5	3	Load base page pointer with 16-bit value
PUSH/POP [DSP][-D]	5	1	EA = BP + DSP - displacement [1 to 8]
PUSHD #literal	6	3	16-bit literal value (high byte first)
PUSHD [PTR]	6	1	Push from 16-bit pointer [0 to 3], high byte first
POPD [PTR]	6	1	Pop to 16-bit pointer [0 to 3], low byte first
PUSH/POP [PTR][TOS]	6	1	EA = (16-bit pointer) + TOS
PUSH/POP [PTR][D]	7	2	EA = (16-bit pointer) + displacement [0 to 255]
PUSH/POP absolute	7	3	Absolute memory address
IN/OUT	7 + 4n	1	Fast I/O instruction, transfer *n* bytes

Table 2.5 ALU Instructions

Mnemonic	Cycles	Size (bytes)	Operation
INC/DEC/NOT	2	1	Increment/decrement/negate TOS
ROLC/RORC	2	1	Rotate left/right TOS through carry
SHL/SHR	2	1	Unsigned left/right shift TOS, clear carry
SHLA/SHRA	2	1	Signed left/right shift TOS into carry
ADD/AND/OR/XOR/ADC	4	1	Operate with NEXT on TOS, drop NEXT
ADD/AND/OR/XOR #literal	3	2	Operate with literal on TOS
(ADD/AND/OR/XOR)_R	7	1	Operate with NEXT on TOS, drop NEXT and return
ALLOC #literal	3	1	Add [1 to 8] to data stack pointer
DEALLOC_R #literal	6	1	Subtract [1 to 8] from data stack pointer and return
SUB NEXT,TOS	4	1	TOS = NEXT - TOS, drop NEXT
SBC NEXT, TOS	4	1	TOS = NEXT - TOS - carry, drop NEXT
SUB TOS,NEXT	4	1	TOS = TOS - NEXT, drop NEXT
XCH	4	1	Exchange TOS and NEXT
INC [PTR]	6	1	Increment 16-bit pointer [0 to 3]

2.2 Memory

2.2.1 Memory Allocation Overview

2.2.1.1 FT 3150 Smart Transceiver Memory Allocation

See Figure 2.5 for a memory map of the FT 3150 Smart Transceiver.

- 512 bytes of in-circuit programmable EEPROM that store the following:
  — Network configuration and addressing information.
  — Unique 48-bit Neuron ID — written at the factory.
  — User-written application code and read-mostly data. See Table 2.6 for available EEPROM space.
- 2,048 bytes of static RAM that store the following:
  — Stack segment, application, and system data.
  — Network and application buffers.

- The processor can access 59,392 bytes of the available 65,536 bytes of memory address space via the external memory interface. The remaining 6,144 bytes of the memory address space are mapped internally.
- 16,384 bytes of the external memory (59,392 bytes total) are required to store the following:

  — The Neuron firmware, including the system firmware executed by the MAC and Network processors, and the executive supporting the application program.
- The rest of the external memory (43,008 bytes) is available for:

  — User-written application code.

  — Additional application read/write and non-volatile data.

  — Additional network buffers and application buffers.

2.2.1.2 FT 3120 Smart Transceiver Memory Allocation

See Figure 2.6 for a memory map of the FT 3120 Smart Transceiver.

- 4,096 bytes of in-circuit programmable EEPROM that store:

  — Network configuration and addressing information.

  — Unique 48-bit Neuron ID — written at the factory.

  — User-written application code and read-mostly data.
- 2,048 bytes of static RAM that store the following:

  — Stack segment, application, and system data.

  — Network buffers and application buffers.
- 12,288 bytes of ROM that store the following:

  — The Neuron firmware, including the system firmware executed by the MAC and network processors, and the executive supporting the application program.

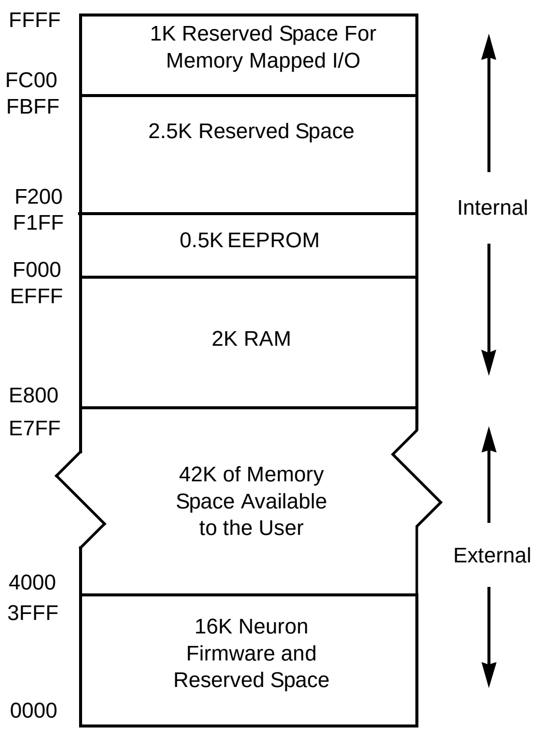

FFFF
1K Reserved Space For Memory Mapped I/O
FC00
FBFF
2.5K Reserved Space
F200
F1FF
0.5K EEPROM
F000
EFFF
2K RAM
E800
E7FF
42K of Memory Space Available to the User
4000
3FFF
16K Neuron Firmware and Reserved Space
0000
Internal
External

Figure 2.5 FT 3150 Smart Transceiver Memory Map

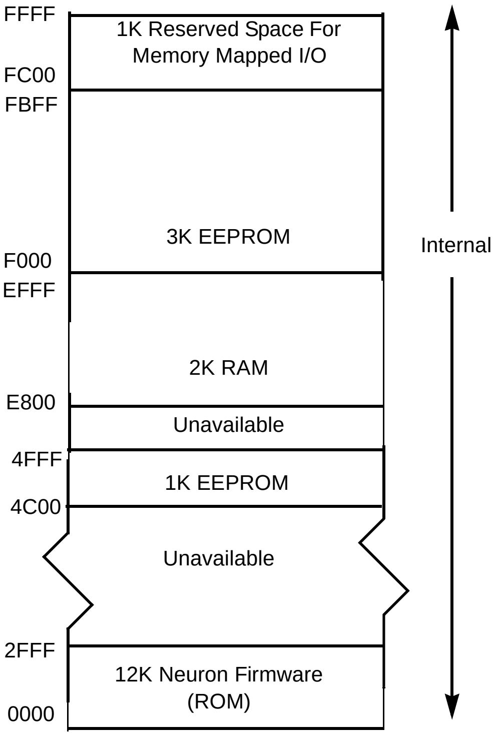


Figure 2.6 FT 3120 Smart Transceiver Memory Map

2.2.2 EEPROM

Both versions of the Smart Transceiver have internal EEPROM containing:

- Network configuration and addressing information.
- Unique 48-bit Neuron ID.
- Optional user-written application code and data tables.

All but 8 bytes of the EEPROM can be written under program control using an on-chip charge pump to generate the required programming voltage. The charge pump operation is transparent to the user. The remaining 8 bytes are written during manufacture, and contain a unique 48-bit identifier for each part called the *Neuron ID*, plus 16 bits for the chip manufacturer's device code. Each byte in the EEPROM region may be written up to 10,000 times. For both the FT 3120 and the FT 3150 Smart Transceivers, the EEPROM stores the installation-specific information such as network addresses and communications parameters. For the FT 3120 Smart Transceiver, the EEPROM also stores the application program generated by the LonBuilder or NodeBuilder development tools. The application code for the FT 3150 Smart Transceiver may be stored either on-chip in the EEPROM memory or off-chip in external memory depending on the size of the application code. See Table 2.6 for available EEPROM space.

For all write operations to the internal EEPROM, the Neuron firmware automatically compares the value in the EEPROM location with the value to be written. If the two are the same, the write operation is not performed. This prevents unnecessary write cycles to the EEPROM, and reduces the average EEPROM write cycle latency.

When the Smart Transceiver is not within the specified power supply voltage range, a pending or on-going EEPROM write is not guaranteed. The Smart Transceiver contains a built-in low-voltage interruption (LVI) circuit that holds the chip in reset when V_{CC} is below a certain voltage. See the *FT 3120 and FT 3150 Smart Transceiver Datasheet* for LVI trip points. This prevents EEPROM data corruption, although in some cases, additional external protection may be appropriate. See section 2.5.2, *RESET Pin*, for more information on LVI circuitry.

In the event of a fault, the on-chip EEPROM of the FT 3150 Smart Transceiver can be reset to its factory default state by executing the EEBLANK program. To do so, program the EEBLANK.NRI file into an external memory device, temporarily replace the application's external ROM or flash with the chip that has EEBLANK.NRI loaded, and power up the device.

After some time, the device's service LED should come on solid, indicating that the EEPROM has been blanked. Then replace the original application ROM or flash. The EEBLANK.NRI file is distributed with the LonBuilder 3.01 (Service Pack 5), NodeBuilder 1.5 (Service Pack 8 or greater), and NodeBuilder 3 (Service Pack 1 or greater) development tools. The file may also be downloaded from the developer's toolbox on Echelon's website at http://www.echelon.com. Versions of EEBLANK.NRI distributed before these Service Packs should not be used with the FT 3150 Smart Transceiver.

The *`set_eeprom_lock()`* function can also be used for additional protection against accidental EEPROM data corruption. This function allows the application program to set the state of the lock on the checksummed portion of the EEPROM. Refer to the *Neuron C Reference Guide* for more information.

The internal EEPROM of a Smart Transceiver will contain a fixed amount of overhead and a network image (configuration), in addition to user code and user data. The following table shows the maximum amount of EEPROM space available for user code and user data assuming a minimally-sized network image. Also shown is the minimum segment size for user data. Constant data is assumed to be part of the code space.

Table 2.6 Memory Usage

Device	Firmware Version	EEPROM Space (Bytes)	Segment Size (Bytes)
FT 3120 Smart Transceiver	13	3969	8
FT 3150 Smart Transceiver	13	384	2

EEPROM must be allocated in increments of the device's segment size, the smallest unit of EEPROM that can be allocated for variable space. For example, if there are three 3-byte variables used, there must be 9 bytes of variable space. For an FT 3120 Smart Transceiver, this would result in the allocation of 16 bytes for variable space, as 16 bytes is the lowest increment of the device segment size (8 bytes) that can store the three 3-byte variables. For an FT 3150 Smart Transceiver, this would result in the allocation of 10 bytes for variable space, as 10 bytes is the lowest increment of the device segment size (2 bytes) that can store the three 3-byte variables.

2.2.3 Static RAM

The FT 3120 and FT 3150 Smart Transceivers contain 2048 bytes of static RAM.

The RAM is used to store the following:

- Stack segment, application, and system data
- Network buffers and application buffers

The RAM state is retained as long as power is applied to the device. After reset, releasing the Smart Transceiver initialization sequence will clear the RAM (see section 2.5.3, *Reset Processes and Timing*, later in this chapter).

2.2.4 Preprogrammed ROM

The FT 3120 Smart Transceiver contains 12,288 bytes of pre-programmed ROM. This memory contains the Neuron firmware, including the LonTalk protocol stack, real time task scheduler, and system function libraries. The Neuron firmware for the FT 3150 Smart Transceiver is stored in external memory. The object code is supplied with the LonBuilder and NodeBuilder tools.

2.2.5 External Memory (FT 3150 Smart Transceiver Only)

This interface supports up to 42Kbytes of external memory space for additional user program and data. The total address space is 64Kbytes. However, the upper 6K of address space is reserved for internal RAM, EEPROM, and memory-mapped I/O (see Figure 2.5 and Figure 2.6), leaving 58K of external address space. Of this space, 16K is used by the Neuron firmware and is reserved for other specific functions. The external memory space can be populated with RAM, ROM, PROM, EPROM, EEPROM, or flash memory in increments of 256 bytes. The memory map for the FT 3150 Smart Transceiver is shown in Figure 2.5. The bus has 8 bidirectional data lines and 16 address lines driven by the processor. Two interface lines ($R/\overline{W}$ and $\overline{E}$) are used for external memory access. Refer to the *FT 3120 and FT 3150 Smart Transceiver Datasheet* for the required access times for the external memory used. If the input clock is scaled down, slower memory can be used. The input clock rates supported by the FT 3150 Smart Transceiver are 20MHz, 10MHz, and 5MHz. The Enable Clock ($\overline{E}$) runs at the system clock rate, which is one-half the input clock rate. All memory, both internal and external, may be accessed by any of the three processors at the appropriate phase of the instruction cycle. Since the instruction cycles of the three processors are offset by one-third of a cycle with respect to each other, the memory bus is used by only one processor at a time.

The *Neuron 3150 Chip External Memory Interface* engineering bulletin provides guidelines for interfacing the FT 3150 Smart Transceiver to different types of memory. A minimum hardware configuration would use one external ROM (PROM or EPROM), containing both the Neuron firmware and user application code. This configuration would **not** allow the system engineer to change the *application code* after installation. The *network image* (*network address* and *connection* information) however, could be altered because this information resides in internal EEPROM. If application downloads over the network are a requirement for maintenance or upgrade and the application code will not fit into the internal EEPROM, then external EEPROM or flash will be necessary. Refer to the *Neuron C Programmer's Guide* for guidelines to reduce code size.

The pins used for external memory interfacing are listed in Table 2.7. The $\overline{E}$ clock signal is used to generate read (or write) signals to external memory. The A15 (address line 15) or a programmable array logic (PAL) decoded signal gated with R/$\overline{W}$ can be used to generate read signals to external memory.

Table 2.7 External Memory Interface Pins

Pin Designation	**Direction**	**Function**
A0 – A15	Output	Address Pins
D0 – D7	Input/Output	Data Pins
$\overline{E}$	Output	Enable Clock
R/$\overline{W}$	Output	Read/Write Select Low

The preferred method of interfacing the Smart Transceiver to another MPU is through the 11 I/O pins using a serial or parallel connection, or through a dual-ported RAM device such as the Cypress CY7C144, CY7C138, or CY7C1342. There are pre-defined serial and parallel I/O models for this purpose which are easily implemented using the Neuron C programming language, or the short stack or MIP firmware can be used to simplify the interface. For more details of dual-ported RAM interfacing, see Appendix B of the *LONWORKS Microprocessor Interface Program User's Guide* (Echelon 078-0017-01).

2.3 Input/Output

2.3.1 Eleven Bidirectional I/O Pins

These pins are usable in several different configurations to provide flexible interfacing to external hardware and access to the internal timer/counters. The logic level of the output pins may be read back by the application processor. See Section 6 for detailed electrical characteristics.

Pins IO4 – IO7 have programmable pull-up current sources. They are enabled or disabled with a compiler directive (see the *Neuron C Reference Guide*). Pins IO0 – IO3 have high current sink capability (20 mA @ 0.8 V). The others have the standard sink capability (1.4 mA @ 0.4 V). All pins (IO0 – IO10) have TTL level inputs with hysteresis. Pins IO0 – IO7 also have low level detect latches.

2.3.2 Two 16-Bit Timer/Counters

The timer/counters are implemented as a load register writable by the processor, a 16-bit counter, and a latch readable by the processor. The 16-bit registers are accessed 1 byte at a time. Both the FT 3150 and FT 3120 Smart Transceivers have one timer/counter whose input is selectable among pins IO4 – IO7, and whose output is pin IO0, and a second timer/counter with input from pin IO4 and output to pin IO1 (Figure 2.7). No I/O pins are dedicated to timer/counter functions. If, for example, Timer/Counter 1 is used for input signals only, then IO0 is available for other input or output functions. Timer/counter clock and enable inputs may be from external pins, or from scaled clocks derived from the system clock; the clock rates of the two timer/counters are independent of each other. External clock actions occur optionally on the rising edge, the falling edge, or both rising and falling edges of the input.

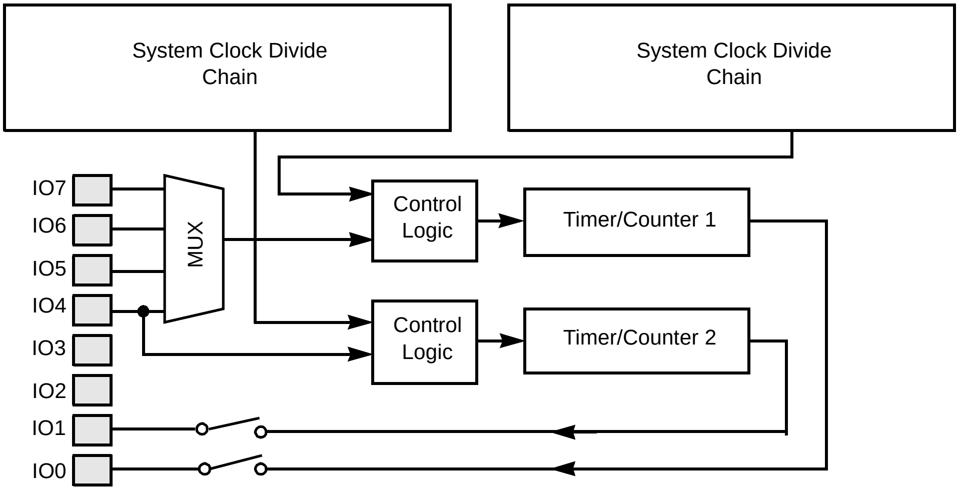


Figure 2.7 Timer/Counter Circuits

2.4 Clock Input

The FT 3120 and FT 3150 Smart Transceivers operate with an input clock of 5, 10, or 20MHz. The FT 3120 Smart Transceiver also supports 40MHz operation. Developer's who are using the LonBuilder 3.0.1 or NodeBuilder 1.5 tools and are upgrading to a clock speed higher than 10MHz should refer to the *readme.txt* file included in the latest Service Pack for the LonBuilder 3.01, or for NodeBuilder 1.5 tools for an in-depth discussion about the software considerations on each platform. The NodeBuilder 3.2 (or later) development tool contains built-in support for these higher clock speeds.

2.4.1 Clock Generation

The Smart Transceiver divides the input clock by a factor of two to provide a symmetrical on-chip system clock. The input clock may be generated either by an external free-running source, or by an on-chip oscillator using an external crystal.

The accuracy of the input clock frequency of the Smart Transceiver must be ±200ppm or better; this requirement can be met with a suitable crystal, but cannot be met with a ceramic resonator.

The Smart Transceiver includes an oscillator that may be used to generate an input clock using an external crystal. For 5 MHz, 10MHz, and 20MHz, either an external clock source or the on-chip crystal oscillator may be used. For 40MHz operation of an FT 3120 Smart Transceiver, an external oscillator must be used.

When an externally generated clock is used to drive the CLK1 CMOS input pin of the Smart Transceiver, CLK2 must be left unconnected or used to drive no more than one external CMOS load. The accuracy of the clock frequency must be ± 0.02% (200 ppm) or better, to ensure that devices may correctly synchronize their bit clocks. Figure 2.8 shows the crystal oscillator circuit. Use the load capacitance and resistor values recommended by the crystal's manufacturer for this circuit. A 60/40 duty cycle or better is required when using an external oscillator as shown in Figure 2.9. An external oscillator must provide CMOS voltage levels to the CLK1 pin.

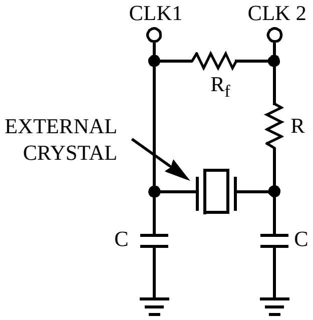


Figure 2.8 Smart Transceiver Clock Generator Circuit

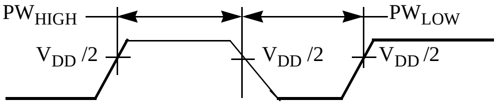


Figure 2.9 Test Point Levels for CLK1 Duty Cycle Measurements

The FT 3120 Smart Transceiver was designed to run at frequencies up to 40MHz using an external clock oscillator. External oscillators generally take several milliseconds to stabilize after power-up. The FT 3120 Smart Transceiver operating at 40MHz must be held in reset until the externally-generated CLK input is stable, so an external power-on-reset-pulse stretching LVI chip/circuit is required. Check the oscillator vendor's specification for more information about startup stabilization times.

2.5 Additional Functions

2.5.1 Reset Function

The reset function is a critical operation in any embedded microcontroller. In the case of theFT 3120 and FT 3150 Smart Transceivers, the reset function plays a key role in the following conditions:

- Initial V_{DD} power up (ensures proper initialization of the Smart Transceiver).
- V_{DD} power fluctuations (manages proper recovery of Smart Transceiver after V_{DD} stabilizes).
- Program recovery (if an application gets lost due to corruption of address or data, an external reset can be used for recovery or the watchdog timer could timeout, causing a watchdog reset).
- V_{DD} power down (ensures proper shut down).
- Helps protect the EEPROM from major corruption.

The Smart Transceivers have four mechanisms to initiate a reset:

- $\overline{\text{RESET}}$ pin is pulled low and then returned high.
- Watchdog timeout occurs during application execution (the timeout period is 210ms at 40MHz; this figure scales inversely with clock frequency).
- Software command either the from the application program or from the network.
- LVI circuit detects a drop in the power supply below a set level.

During any of the reset functions, when the $\overline{\text{RESET}}$ pin is in the low state, the Smart Transceiver pins go to the states described in the list below. Figure 2.11 also illustrates the condition of the pins during reset and the Smart Transceivers initialization sequence after reset is returned high again.

- Oscillator continues to run
- All processor functions stop
- $\overline{\text{SERVICE}}$ pin goes to high impedance
- I/O pins go to high impedance
- All output address pins go to 0xFFFF (FT 3150 Smart Transceiver only)
- All data pins become outputs with high or low states (FT 3150 Smart Transceiver only)
- $\overline{\text{E}}$ clock goes high (FT 3150 Smart Transceiver only)
- R/$\overline{\text{W}}$ goes low (FT 3150 Smart Transceiver only)

When the $\overline{\text{RESET}}$ pin is released back to a high state, the Smart Transceiver begins its initialization procedure starting at address 0x0001. The time it takes the Smart Transceiver to complete its initialization differs between Smart Transceivers, the different firmware versions that are being run, and the memory space used by the application (code and data). This will be discussed later in this section.

2.5.2 $\overline{\text{RESET}}$ Pin

The $\overline{\text{RESET}}$ pin is both an input and an output. As an input, the $\overline{\text{RESET}}$ pin is internally pulled high by a current source acting as a pull-up resistor. The $\overline{\text{RESET}}$ pin becomes an output when any of the following events occur:

- Watchdog Timer event.
- Software reset initialization.
- Internal LVI detects a low voltage.
- $\overline{\text{RESET}}$ pin drops below the internal trip point.

2.5.2.1 Power Up Sequence

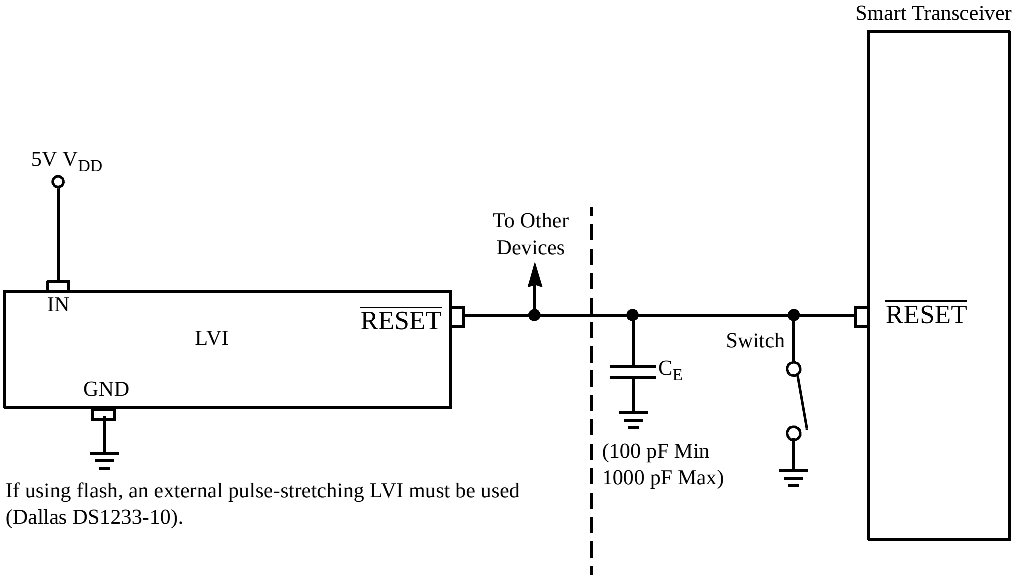
During power up sequences, the $\overline{\text{RESET}}$ pin should be held low until the power supply is stable, to prevent start-up malfunctioning. Likewise, when powering down, the Smart Transceiver $\overline{\text{RESET}}$ pin should go to a low state before the power supply goes below the minimum operating voltage of the Smart Transceiver.

WARNING: If proper reset recovery circuitry is not used, the Smart Transceiver can go applicationless or unconfigured. The applicationless or unconfigured state occurs when the checksum error verification routine detects a corruption in memory which could have falsely been detected due to improper reset sequence or noise on the power supply. Several options exist in the LonBuilder and NodeBuilder tools to allow a reboot on checksum failure.

Figure 2.11 shows typical external $\overline{\text{RESET}}$ components. The total capacitance directly connected to the $\overline{\text{RESET}}$ pin, including stray and external device input capacitance, must not exceed 1000 pF. This ensures that the Smart Transceiver can successfully output a reset down to below 0.8V. The 100 pF minimum capacitance is required for noise immunity.

2.5.2.2 Software Controlled Reset

When the CPU watchdog timer expires, or a software command to reset occurs, the $\overline{\text{RESET}}$ pin is pulled low for 256 CLK1 clock cycles. The $\overline{\text{RESET}}$ pin and external capacitor (100 ≤ x ≤ 1000 pF) are allowed to begin charging and provide the required duration of reset.



If using flash, an external pulse-stretching LVI must be used (Dallas DS1233-10).

Figure 2.10 Example of a Reset Circuit

2.5.2.3 Watchdog Timer

The Smart Transceivers are protected against malfunctioning software or memory faults by three watchdog timers, one for each processor that makes up the Neuron core. If application or system software fails to reset these timers periodically, the entire Smart Transceiver is automatically reset. The watchdog period is approximately 210 ms at a 40MHz input clock rate and scales inversely with the input clock rate.

2.5.2.4 LVI Considerations

The FT 3120 and FT 3150 Smart Transceivers include an internal LVI to ensure that they only operate above the minimum voltage threshold. See the *FT 3120 and FT 3150 Smart Transceiver Datasheet* for LVI trip points. If the circuit operates below this voltage, improper operation could occur. For example, if the Smart Transceiver is writing to an internal or external EEPROM or to flash memory when a reset event is initiated, then that data could be corrupted.

When using external flash memory for the FT 3150 Smart Transceiver device, an external pulse-stretching LVI of greater than 50 ms should be used (Echelon recommends using Dallas Semiconductor Part No. DS1233-5). When using an external oscillator to drive the CLK1 pin of the FT 3150 or FT 3120 Smart Transceivers, a power-on-pulse-stretching LVI may be needed to ensure that the external oscillator has stabilized before the Smart Transceiver is released from reset.

Since the $\overline{\text{RESET}}$ pin of the Smart Transceiver is bidirectional, an external LVI must have an open-drain or open-collector output. If an external LVI actively drives the $\overline{\text{RESET}}$ pin high, then the Smart Transceiver will not be able to reliably assert the $\overline{\text{RESET}}$ pin (low) during internal resets. This contention on the Smart Transceiver $\overline{\text{RESET}}$ pin can cause anomalous behavior, from applicationless errors to physical damage to the Smart Transceiver reset circuitry.

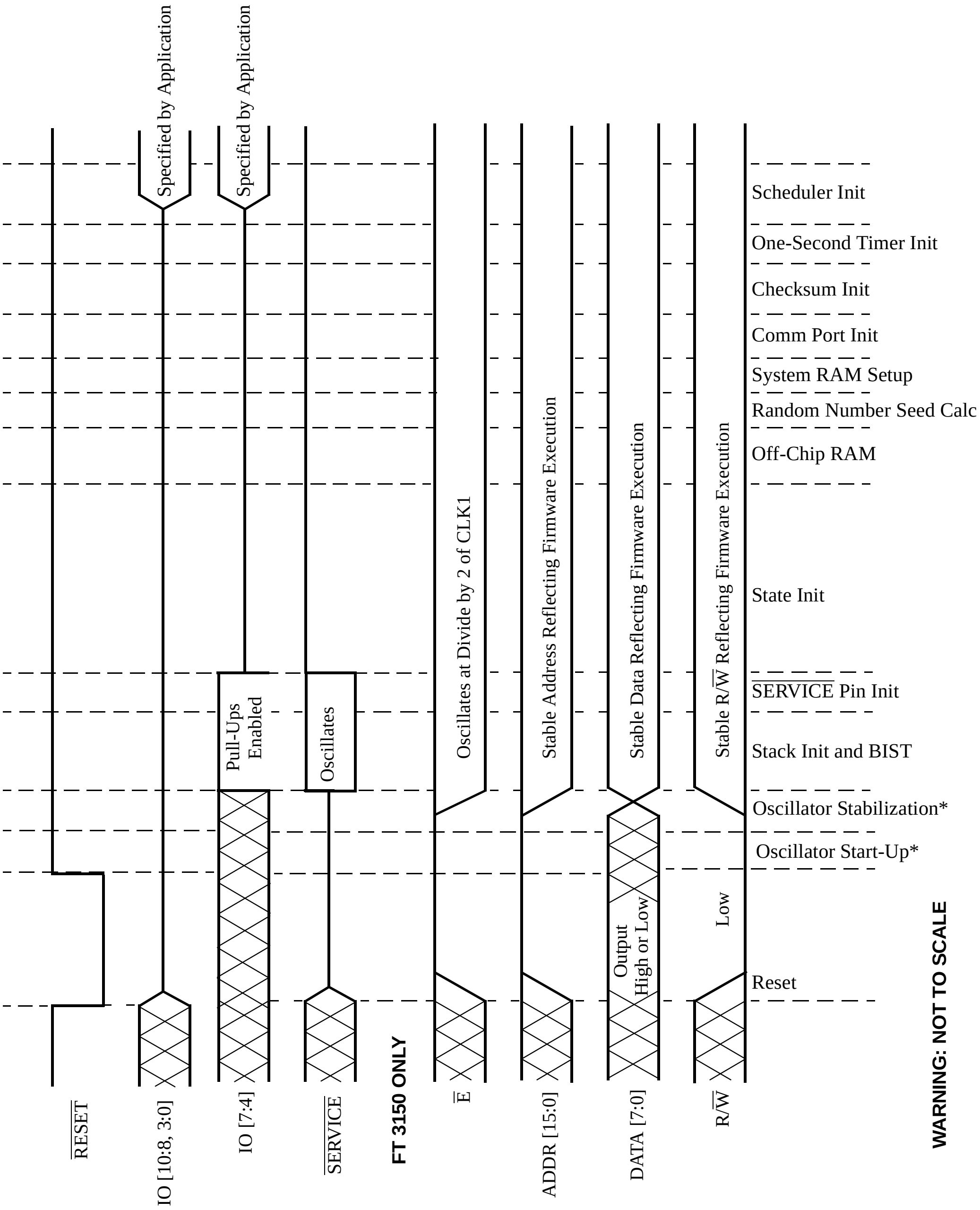
2.5.3 Reset Processes and Timing

During the reset period, the I/O pins are in a high-impedance state. The FT 3150 Smart Transceiver address lines A15 – A0 are forced to 0xFFFF, R/$\overline{\text{W}}$ is forced to 0, and $\overline{\text{E}}$ is forced to 1. The data lines are undetermined but driven high or low, so they will not float and draw excess current. The $\overline{\text{SERVICE}}$ pin is high impedance during reset. Reset overrides the effect of $\overline{\text{E}}$ clock on data lines in that, in normal operations the data bus is only driven in a write cycle during the $\overline{\text{E}}$ clock low portion of the bus cycle, while reset forces the data bus to be driven. The steps followed in preparing the Smart Transceiver to execute the application code are discussed below. These steps are summarized in Figure 2.11.

After the $\overline{\text{RESET}}$ pin is released, the Smart Transceiver performs hardware and firmware initialization before executing application programs. These tasks are:

- Oscillator start-up
- Oscillator stabilization
- Stack initialization and built-in self-test (BIST)

- $\overline{\text{SERVICE}}$ pin initialization
- State initialization



*NOTE: On power up, the oscillator will start running before $\overline{\text{RESET}}$ is released.

Figure 2.11 RESET Timeline for FT 3120 and FT 3150 Smart Transceivers

- Off-chip RAM initialization
- Random number seed calculation
- System RAM setup
- Communication port initialization
- Checksum initialization
- One-second timer initialization
- Scheduler initialization

During internal oscillator start up (after power up), the Smart Transceiver waits for the oscillator signal amplitude to grow before using the oscillator waveform as the system clock. This period depends on the type of oscillator used and its frequency, and begins as soon as power is applied to the oscillator and is independent of the $\overline{\text{RESET}}$ pin. The oscillator start-up period may end before or after $\overline{\text{RESET}}$ is released, depending on the duration of reset and the time required by the oscillator to start up.

After the oscillator has started up, the Smart Transceiver counts additional transitions on CLK1 to allow the oscillator's frequency to stabilize. From the time $\overline{\text{RESET}}$ is asserted until the end of the oscillator stabilization period, the I/O pins are in a high-impedance state. The $\overline{\text{E}}$ signal goes inactive (high) immediately after reset goes low, and the address bus becomes high (0xFFFF) to deselect external devices.

The stack initialization and BIST task tests the on-chip RAM, the timer/counter logic, and the counter logic. For the test to pass, all three processors and the ROM must be functioning. A flag is set to indicate whether the Smart Transceiver passed or failed the BIST. The RAM is cleared to all 0s by the end of this step. At the beginning of this task, the pull-ups on IO[7:4] are enabled, so that a weak high state can be observed on these pins. The $\overline{\text{SERVICE}}$ pin oscillates between a solid low and a weak high. The memory interface signals reflect execution of these tasks.

If the RAM self-test fails, the device goes offline, the service LED comes on solid, and an error is logged in the device's status structure.

Self-test results are available in the first byte of RAM (0xE800) as follows:

Value	**Description**
0	No Failure
1	RAM failure
2	Timer/counter failure
3	Counter failure
4	Configured input clock rate exceeds the chip maximum

The $\overline{\text{SERVICE}}$ pin initialization task turns off the $\overline{\text{SERVICE}}$ pin (high state).

The state initialization task determines if a Smart Transceiver boot is required (FT 3150 Smart Transceiver only), and performs the boot if it is required. The Smart Transceiver decides to perform a boot if it is blank, or if the boot ID does not match the boot ID in ROM.

The off-chip RAM initialization task checks the memory map to determine if any off-chip RAM is present and then either tests and clears all of the off-chip RAM or, optionally, clears the application RAM area only. This choice is controlled by the application program via a Neuron C compiler directive. This task applies only to the FT 3150 Smart Transceiver.

The random number seed calculation task creates a seed for the random number generator.

The system RAM setup task sets up internal system pointers as well as the linked lists of system buffers.

The checksum initialization task generates or checks the checksums of the nonvolatile writable memories. If the boot process was executed for the configured or unconfigured states, in the state initialization task, then the checksums are generated; otherwise, they are checked. This process includes on-chip EEPROM, off-chip EEPROM, flash, and off-chip nonvolatile RAM. There are two checksums, one for the configuration image and one for the application image. In each case, the checksum is a negated two’s complement sum of the values in the image.

The one-second timer initialization task initializes the one-second timer. At this point, the network processor is available to accept incoming packets.

The scheduler initialization task allows the application processor to perform application-related initialization as follows:

- State wait — wait for the device to leave the applicationless state.
- Pointer initialization — perform a global pointer initialization.
- Initialization step — execute initialization task, which is created by the compiler/linker to handle initialization of static variables and the timer/counters.
- I/O pin initialization step — initialize I/O pins based on application definition. Prior to this point, I/O pins are high impedance.
- State wait II — wait for the device to leave the unconfigured or hard-offline state. If waiting was required, a flag is set to indicate that the device should come up offline.
- Parallel I/O synchronization — devices using parallel I/O attempt to execute the master/slave synchronization protocol at this point.
- Reset task — execute the application reset task (when (reset{ })).
- If the offline flag was set, go offline and execute the offline task. If the BIST flag indicated a failure, then the $\overline{\text{SERVICE}}$ pin is turned on and the offline task is executed. Otherwise, the scheduler starts its normal task scheduling loop.

The amount of time required to perform these steps depends on many factors, including: Smart Transceiver model; input clock rate; whether or not the device performs a boot process; whether the device is applicationless, configured, or unconfigured; amount of off-chip RAM; whether the off-chip RAM is tested or simply cleared; the number of buffers allocated; and application initialization. Table 2.8 and Table 2.9 summarize the number of input clock cycles (CLK1) required for each of these steps for the FT 3120 and the FT 3150 Smart Transceivers. The times are approximate and are given as functions of the most significant application variables.

Table 2.8 FT 3120 Smart Transceiver Reset Sequence Time

Step	**Number of CLK1 Cycles**	**Notes**
Stack Initialization and BIST	386,000	
$\overline{\text{SERVICE}}$ Pin Initialization	1000	
State Initialization	250 (for no boot) 2,275,000 (for boot)	
Off-Chip RAM Initialization	0	
Random Number Seed Calculation	0	1
System RAM Set-up	21,000 + 600*B	2
Communication Port Initialization	0	1
Checksum Initialization	3400 + 175*M	3
One-Second Timer Initialization	6100	
Scheduler Initialization	≥ 7400	4

Notes:

Note 1) These tasks run in parallel with other tasks.

Note 2) B is the number of application and/or network buffers allocated.

Note 3) M is the number of bytes to be checksummed.

Note 4) Assumes a trivial initialization task, no reset task and the configured state.

For example, the timing of each of these steps is shown for a FT 3120 Smart Transceiver application with the following parameters: 10MHz input clock, crystal oscillator, no boot required, at least 10 application and/or network buffers, and 500 bytes of EEPROM checksummed.

Stack Initialization and BIST	38.6 ms	
$\overline{\text{SERVICE}}$ Pin Initialization	0.1 ms	
State Initialization	0.025 ms	
Off-Chip RAM Initialization	0 ms	
Random Number Seed Calculation	0 ms	
System RAM Setup	2.7 ms	
Communication Port Initialization	0 ms	
Checksum Initialization	10.8 ms	
One-Second Timer Initialization	0.61 ms	
Scheduler Initialization	0.74 ms	
Total		53.7 ms

Table 2.9 FT 3150 Smart Transceiver Reset Sequence Time

Step	**Number of CLK1 Cycles**	**Notes**
Stack Initialization and BIST	425,000	
$\overline{\text{SERVICE}}$ Pin Initialization	1000	
State Initialization	1300 (for no boot) 70,000 + 25 ms*E (for boot)	1
Off-Chip RAM Initialization	24,000 + 214*R (for test and clear) 24,000 + 152*R_a (for clear only)	2 3
Random Number Seed Calculation	50,000 max	
System RAM Setup	27,000 + 1500*B	4
Communication Port Initialization	0	5
Checksum Initialization	7200 + 175*M (for no boot) 82,000 + 100 ms + 175*M (for boot)	6, 7
One-Second Timer Initialization	6100	
Scheduler Initialization	≥ 7400	8

Notes:

Note 1) E is the number of non-zero bytes being written (ranges from 10 to 504).

Note 2) R is the number of off-chip RAM bytes.

Note 3) R_a is the number of non-system off-chip RAM bytes.

Note 4) B is the number of application and/or network buffers allocated.

Note 5) These tasks run in parallel with other tasks.

Note 6) M is the number of bytes to be checksummed.

Note 7) Only if booting to the configured or unconfigured state; if booting to the applicationless state, use the "no boot" equation.

Note 8) Assumes a trivial initialization task, no reset task, and the configured state.

For example, the timing of each of these steps is shown for a FT 3150 Smart Transceiver application with the following parameters: 10MHz input clock, crystal oscillator, no boot required, 16K external RAM, test and clear external RAM, at least 10 application and/or network buffers, and 500 bytes of EEPROM checksummed.

Stack Initialization and BIST	42.5 ms	
SERVICE Pin Initialization	0.1 ms	
State Initialization	0.13 ms	
Off-Chip RAM Initialization	353 ms	
Random Number Seed Calculation	5 ms	
System RAM Setup	4.2 ms	
Communication Port Initialization	0 ms	
Checksum Initialization	12.5 ms	
One-Second Timer Initialization	0.61 ms	
Scheduler Initialization	0.74 ms	
Total		418 ms

Use the following compiler directive to disable testing of off-chip RAM:

```
# pragma ram_test_off
```

2.5.4 SERVICE Pin

The SERVICE pin alternates between input and open-drain output at a 76 Hz rate with a 50% duty cycle. When it is an output, it can sink 20 mA for use in driving a LED. When it is used exclusively as an input, it has an optional on-chip pull-up to bring the input to an inactive-high state for use when the LED and pull-up resistor are not connected. Under control of the Neuron firmware, this pin is used during configuration, installation, and maintenance of the device containing the Smart Transceiver. The firmware flashes the LED at a 1/2-Hz rate when the Smart Transceiver has not been configured with network address information. Grounding the SERVICE pin causes the Smart Transceiver to transmit a network management message containing its unique 48-bit Neuron ID and the application's program ID on the network. This information may then be used by a network tool to install and configure the device. A typical circuit for the SERVICE pin LED and push-button is shown in Figure 2.12. During reset the $\overline{\text{SERVICE}}$ pin state is indeterminate. The default state of the $\overline{\text{SERVICE}}$ pin pull-up is enabled.

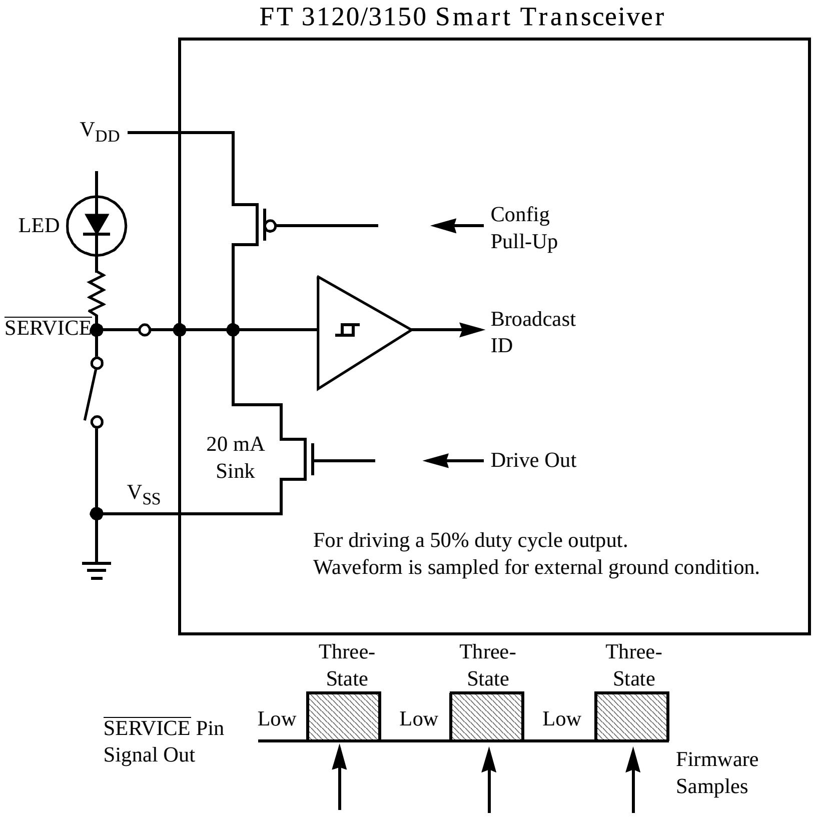


Figure 2.12 Smart Transceiver $\overline{\text{SERVICE}}$ Pin Circuit

Table 2.10 Service LED Behavior During Different States

Device State	0xF015 State Code	Service LED
Applicationless and Unconfigured	3	On
Unconfigured (but with an Application)	2	Flashing
Configured, Hard Offline	6	Off
Configured	4	Off
3150 Defective External Memory	—	On

The $\overline{\text{SERVICE}}$ pin is active low and the service pin message is sent once maximum per $\overline{\text{SERVICE}}$ pin transition. The service pin message goes into the next available priority or non-priority output network buffer.

2.6 Integrity Mechanisms

2.6.1 Memory Integrity Using Checksums

To ensure the integrity of the Smart Transceiver's memory, the Neuron firmware maintains a number of checksums. Each checksum is a single byte and is the two's complement of the sum of all bytes it covers. These checksums are verified during reset processing and also on a continual basis via a background diagnostic process. There are three main checksums used to verify the integrity of the Smart Transceiver's memory:

- Configuration image checksum
- Application image checksum
- System image checksum (off-chip system image only)

The configuration image checksum covers the network configuration information and communication parameters residing in the on-chip EEPROM. The default behavior is that a configuration checksum error causes the device to go to the unconfigured state. Refer to Table 2.12 for other options.

The application image checksum covers the application code in both on-chip EEPROM and any application code in off-chip EEPROM, NVRAM, or flash memory. This checksum can optionally be extended to cover any application code in off-chip ROM as well. The default behavior is that an application checksum error causes the device to go to the applicationless state. Application read/write data residing in EEPROM, NVRAM, or flash is not checksummed. Refer to Table 2.12 for other options.

Table 2.11 Checksum Coverage of Smart Transceiver Memory Areas

Memory Area	**Checksum**
System image (optionally covered by application checksum on the FT 3150)	System
Any off-chip ROM code (optionally covered by Application checksum on the FT 3150)	Application
Any off-chip flash, EEPROM, or NVRAM code	Application
Any off-chip RAM code	Application
Configuration image	Configuration
All on-chip EEPROM code	Application

In the FT 3150 Smart Transceiver, all memory areas listed in Figure 2.11 except for on-chip EEPROM code have their own checksum so that checksum errors can be further isolated. An unconfigured or configured device continually checks its application checksum in the background at the rate of 1 byte per iteration through the network processor's main loop (3 bytes per millisecond when running at 10MHz with no network activity).

The system image checksum covers the system image. It is only available when the system image resides in off-chip memory and its use is optional. A system image checksum error always forces the device to the applicationless state.

No checksum is computed if the device is in the applicationless state.

The checksums are all verified during reset processing by the network processor and as part of the background diagnostic process. The background diagnostic process causes the device to reset when an error is detected; no state change occurs. It is assumed that any persistent error will be found by the reset processing.

Upon detecting a checksum error, the reset process will force the appropriate state and log an error in the error log. For the FT 3150 Smart Transceiver, a checksum must fail twice during reset processing in order for it to be deemed bad.

2.6.2 Reboot and Integrity Options Word

An FT 3150 Smart Transceiver has a number of options for actions taken following a checksum error or other memory related fatal errors. The 16-bit word resides in the system image and is defined as part of the device’s export options in the LonBuilder and NodeBuilder tools.

The recovery process relies on the fact that the initial on-chip EEPROM image for the application, configuration, and communication parameter data reside in the off-chip system image. During initial power up, the system image data is copied (booted) to on-chip EEPROM. The recovery process recopies or reboots the suspect areas as dictated by the error and the recovery options. **Any changes made to the on-chip EEPROM (e.g., a network application load or network tool initiated reconfiguration) after the initial boot are lost in the recovery process.** The recovery action is defined by setting a combination of bits as defined by the following bit masks (Table 2.12).

Table 2.12 Recovery Action Bit Masks

Recovery Word	**Description**
0x0001	Reboot application if application fatal error.
0x0002	Always reboot application on reset (see NOTE 1).
0x0004	Reboot configuration if configuration checksum fails.
0x0008	Reboot configuration on an application fatal error.
0x0010	Always reboot configuration on reset.
0x0020	Reboot communication parameters if configuration checksum fails.
0x0040	Reboot communication parameters if type or rate mismatch.
0x0080	Always reboot communication parameters on reset.
0x0100	Reboot EEPROM variables when rebooting application.
0x0200	Applicationless state is considered to be an application fatal error. If option 0x0001 or 0x0008 is set, applicationless state will result in a reboot. Application fatal errors are defined below (see NOTE 1).
0x0400	Checksum all code, including system image.

NOTE 1: Applications exported with these options cannot be loaded over the network.

In the above options, “configuration” does not include the communication parameters since their recovery is governed separately. Also, fatal application errors refer to application image checksum errors, memory allocation failures, and memory map failures. Refer to *Programming 3150 Chip Memory* in the *LonBuilder User’s Guide* (Revision 3.0) or to *Loading an Application Image* in the *NodeBuilder User’s Guide* (Release 3 Revision 2) for more information.

The configuration will be rebooted independently of the application only if all the configuration table sizes match between EEPROM and ROM. This avoids a situation where a new application with different table sizes is loaded over the network, and a reboot of the configuration corrupts the program.

When an EEPROM recovery occurs due to a checksum failure or other error, the event will be logged in the Smart Transceiver's error table. A test command will show *EEPROM recovery occurred* as the last error logged.

2.6.3 Reset Processing

During reset processing, the configuration checksum is checked first. If bad, and no configuration recovery options are set, then a configuration checksum error is logged, the checksum repaired, and the device state is changed to unconfigured. If the configuration recovery option is set, the configuration is recovered.

Next, the application checksum is checked. If bad, and the checksum error is in the system image, then a system image checksum error is logged and the device state is changed to applicationless.

If the application checksum is bad, and no application recovery options are set, an application checksum error is logged and the device state is changed to applicationless.

If the application checksum is bad and an application recovery option is set and the boot application does not contain references to any off-chip ROM, flash, EEPROM, NVRAM, or RAM code, or there are no checksum errors in any of these regions, then the application is recovered. Otherwise, an application checksum error is logged and the device goes applicationless.

2.6.4 Signatures

All off-chip code areas have a 2-byte cyclic redundancy check (CRC) called the signature, immediately following the area checksum. Signatures are stored in the area and in the memory map. Mismatches between the area signature and memory map copy of the signature result in the device going applicationless. This mechanism prevents a partial application load over the network which is incompatible with any unloaded code (such as code in ROM).

3

Input/Output Interfaces

The FT 3120 and FT 3150 Smart Transceivers connect to application-specific external hardware via 11 pins, named IO0-IO10. These pins may be configured in numerous ways to provide flexible input and output functions with minimal external circuitry. The programming model (Neuron C language) allows the programmer to declare one or more pins as I/O objects. An I/O object provides programmable access to an I/O driver for a specified on-chip I/O hardware configuration and a specified input or output waveform definition. The user's program may then refer to these objects in *io_in* and *io_out*() system calls to perform the actual input/output function during execution of the program. Certain events are associated with changes in input values. The task scheduler can thus execute associated application code when these changes occur.

There are 34 different I/O objects available for use with the Smart Transceivers. Most I/O Objects are available in the FT 3150 and FT 3120 Smart Transceiver system images by default. If an object not included in the default system image is required by an application, the development tool will link the appropriate objects into available memory space. For FT 3120 Smart Transceiver designs, this means that internal EEPROM space must be used for the additional object. For FT 3150 Smart Transceiver designs, the object will be added to an external flash or ROM region beyond the 16KB space reserved for the system image.

The FT 3120 and FT 3150 Smart Transceivers have two 16-bit timer/counters on-chip (see Figure 2.7). The input to timer/counter 1, also called the *multiplexed timer/counter*, is selectable among pins IO4 – IO7, via a programmable multiplexer (mux) and its output may be connected to pin IO0. The input to timer/counter 2, also called the *dedicated timer/counter*, may be connected to pin IO4 and its output to pin IO1. The timer/counters are implemented as a 16-bit load register writable by the CPU, a 16-bit counter, and a 16-bit latch readable by the CPU. The load register and latch are accessed a byte at a time. No I/O pins are dedicated to timer/counter functions. If, for example, timer/counter 1 is used for input signals only, then IO0 is available for other input or output functions. Timer/counter clock and enable inputs may be from external pins, or from scaled clocks derived from the system clock; the clock rates of the two timer/counters are independent of each other. External clock actions occur optionally on the rising edge, the falling edge, or both rising and falling edges of the input.

Multiple timer/counter input objects may be declared on different pins within a single application. By calling the *io_select()* function, the application can use the first timer/counter to implement up to four different input objects. If a timer/counter is configured to implement one of the output objects, or is configured as a quadrature input object, then it can not be reassigned to another timer/counter object in the same application program.

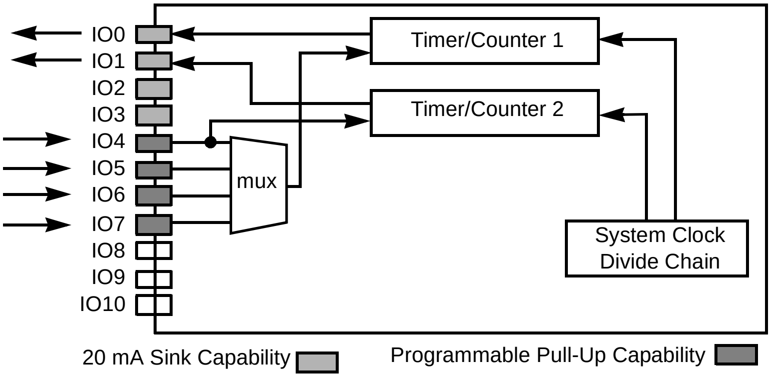


Figure 3.1 Smart Transceiver Timer/Counter External Connections

3.1 Hardware Considerations

Tables 3.1 through 3.5 list the 34 different I/O objects available. Various I/O objects of different types may be used simultaneously. Figure 3.3 summarizes the pin configuration for each of the I/O objects. For the electrical characteristics of these pins, refer to the *FT 3120 and FT 3150 Smart Transceiver Datasheet*. The following sections contain detailed descriptions of all the I/O objects. The application program may optionally specify the initial values of digital outputs. Pins configured as outputs may also be read as inputs, returning the value last written. Pins IO4 – IO7 have optional pull-up current sources that act like pull-up resistors (see Figure 3.1). These are enabled with a Neuron C compiler directive (#pragma enable_io_pullups). Pins IO0 – IO3 have high sink capability. The others have standard sink capability. Pins IO0 – IO7 have low-level detect latches. **The latency and timing values described later in this section are typical at 10MHz.** The accuracy of these values is ± 10%. Most latency values scale down at higher input clock rates and scale up at lower input clock rates.

The I/O pull-ups are always enabled during the stack initialization and BIST task. This can cause a problem in some applications, for example driving a relay. The best solution is to use an I/O that does not have a pull-up. However, if an I/O with a pull-up must be used, a pull-down resistor could be used to overcome the effects of the pull-up. Typically, a pull-down in the range of 2.4kΩ to 2.7kΩ is adequate.

Table 3.1 Summary of Direct I/O Objects

I/O Object	Applicable I/O Pins	Input/Output Value	Page No.
Bit Input	IO0 – IO10	0, 1 binary data	3-12
Bit Output	IO0 – IO10	0, 1 binary data	3-12
Byte Input	IO0 – IO7	0 – 255 binary data	3-14
Byte Output	IO0 – IO7	0 – 255 binary data	3-14
Leveldetect Input	IO0 – IO7	Logic 0 level detected	3-15
Nibble Input	Any adjacent 4 in IO0 – IO7	0 – 15 binary data	3-15
Nibble Output	Any adjacent 4 in IO0 – IO7	0 – 15 binary data	3-15

Table 3.2 Summary of Parallel I/O Objects

I/O Object	Applicable I/O Pins	Input/Output Value	Page No.
Muxbus I/O	IO0 – IO10	Parallel bidirectional port using multiplexed addressing	3-18
Parallel I/O	IO0 – IO10	Parallel bidirectional handshaking port	3-20

Table 3.3 Summary of Serial I/O Objects

I/O Object	Applicable I/O Pins	Input/Output Value	Page No.
Bitshift Input	Any adjacent pair (except IO7 + IO8)	Up to 16 bits of clocked data	3-33
Bitshift Output	Any adjacent pair (except IO7 + IO8)	Up to 16 bits of clocked data	3-33
I^2C	IO8 + IO9	Up to 255 bytes of bidirectional serial data	3-36
Magcard Input	IO8 + IO9 + (one of IO0 – IO7)	Encoded ISO7811 track 2 data stream from a magnetic card reader	3-38
Magtrack1	IO8 + IO9 + (one of IO0 – IO7)	Encoded ISO3554 track 1 data stream from a magnetic card reader	3-40
Neurowire I/O	IO8 + IO9 + IO10 + (one of IO0 – IO7)	Up to 256 bits of bidirectional serial data	3-42
Serial Input	IO8	8-bit characters	3-46
Serial Output	IO10	8-bit characters	3-46
Touch I/O	IO0 – IO7	Up to 2048 bits of input or output bits	3-48
Wiegand Input	Any adjacent pair in IO0 – IO7	Encoded data stream from Wiegand card reader	3-51

Table 3.4 Summary of Timer/Counter Input Objects

I/O Object	Applicable I/O Pins	Input Signal	Page No.
Dualslope Input	IO0, IO1 + (one of IO4 – IO7)	Comparator output of the dualslope converter logic	3-54
Edgelog Input	IO4	A stream of input transitions	3-56
Infrared Input	IO4 – IO7	Encoded data stream from an infrared demodulator	3-58
Ontime Input	IO4 – IO7	Pulse width of 0.2 µs – 1.678 s	3-59
Period Input	IO4 – IO7	Signal period of 0.2 µs – 1.678 s	3-60
Pulsecount Input	IO4 – IO7	0 – 65,535 input edges during 0.839 s	3-62
Quadrature Input	IO4 + IO5, IO6 + IO7	± 16,383 binary Gray code transitions	3-64
Totalcount Input	IO4 – IO7	0 – 65,535 input edges	3-66

Table 3.5 Summary of Timer/Counter Output Objects

I/O Object	Applicable I/O Pins	Output Signal	Page No.
Edgedivide Output	IO0, IO1 + (one of IO4 – IO7)	Output frequency is the input frequency divided by a user-specified number	3-68
Frequency Output	IO0, IO1	Square wave of 0.3 Hz to 2.5MHz	3-70
Oneshot Output	IO0, IO1	Pulse of duration 0.2 µs to 1.678 s	3-71
Pulsecount Output	IO0, IO1	0 – 65,535 pulses	3-73
Pulsewidth Output	IO0, IO1	0 – 100% duty cycle pulse train	3-75
Triac Output	IO0, IO1 + (one of IO4 – IO7)	Delay of output pulse with respect to input edge	3-77
Triggered-count Output	IO0, IO1 + (one of IO4 – IO7)	Output pulse controlled by counting input edges	3-79

To maintain and provide consistent behavior for external events and to prevent metastability, all 11 I/O pins of the Smart Transceiver, when configured as inputs, are passed through a hardware synchronization block sampled by the internal system clock. This is always the input clock divided by two (e.g. 10MHz ÷ 2 = 5MHz). For any signal to be reliably synchronized with a 10MHz input clock, it must be at least 220 ns in duration (see Figure 3.2).

All inputs are software sampled during *when* statement processing. The latency in sampling is dependent on the I/O object which is being executed (see I/O timing specification and *Neuron C Programmer's Guide* for more information). These latency values scale inversely with the input clock. Thus, any event that lasts longer than 220 ns will be synchronized by hardware, but there will be latency in software sampling resulting in a delay detecting the event. If the state changes at a faster rate than software sampling can occur, then the interim changes will go undetected.

There are two exceptions to the synchronization block. First, the chip select ($\overline{\text{CS}}$) input used in the slave B mode of the parallel I/O object; this input will recognize rising edges asynchronously (see 3-18). Second, the leveldetect input is latched by a flip flop with a 200ns clock. The leveldetect transition event will be latched, but there will be a delay in software detection (see 3-15). The input timer/counter functions are also different, in that events on the I/O pins will be accurately measured and a value returned to a register, regardless of the state of the application processor. However, the application processor may be delayed in reading the register. Consult the *Neuron C Programmer's Guide* for detailed programming information.

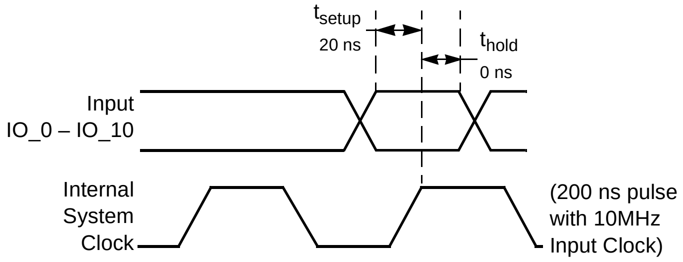
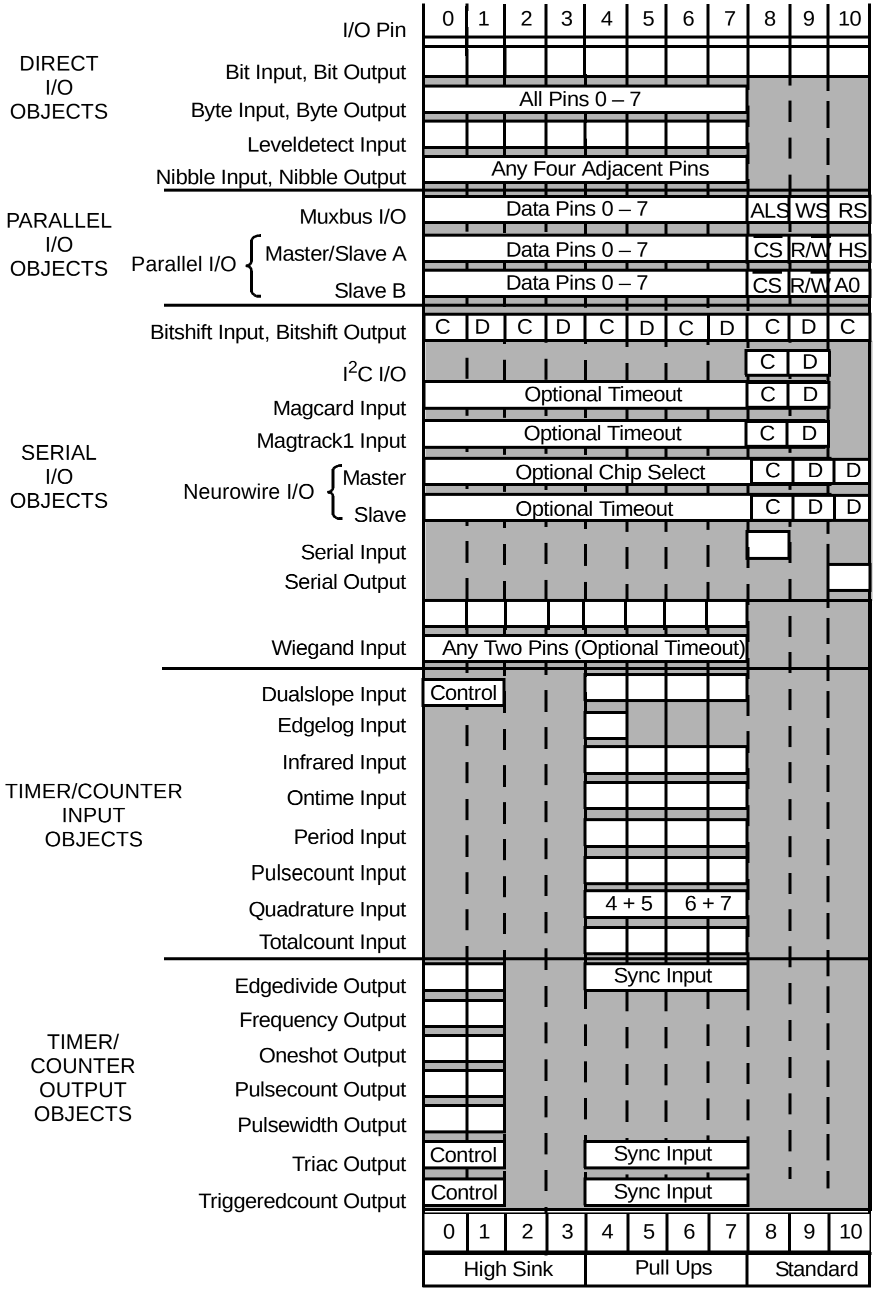


Figure 3.2 Synchronization of External Signals



Notes:

C = Clock, D = Data

Bitshift, I^{2}C, Magcard, Magtrack, Neurowire

Timer/Counter 1 Devices

One of:

IO_6 input quadrature

IO_4 input edgelog

IO_0 output [triac | triggeredcount | edgedivide] sync(IO_4..7)

IO_0 output [frequency | oneshot | pulsecount | pulsewidth]

Or up to four of:

IO_4 input [ontime | period | pulsecount | totalcount | dualslope | infrared] mux

IO_5..7 input [ontime | period | pulsecount | totalcount | dualslope | infrared]

Timer/Counter 2 Devices

One of:

IO_4 input quadrature

IO_4 input edgelog

IO_1 output [triac | triggeredcount | edgedivide] sync(IO_4)

IO_1 output [frequency | oneshot | pulsecount | pulsewidth]

IO_4 input [ontime | period | pulsecount | totalcount | dualslope | infrared] ded

Figure 3.3 Summary of I/O Objects

3.2 I/O Timing Issues

The Smart Transceiver I/O timing is influenced by three separate, yet overlapping areas of the overall chip architecture:

- The scheduler
- The I/O object's firmware
- The Smart Transceiver hardware

The contribution of the scheduler to the overall timing characteristic is approximately uniform across all 34 I/O function blocks since its contribution to the overall I/O timing is at a relatively high functional level.

The contribution of firmware and hardware varies from one I/O object to another (e.g., Bit I/O versus Neurowire I/O), with one area generally being the dominant factor.

3.2.1 Scheduler-Related I/O Timing Information

As part of the Smart Transceiver firmware, the scheduler provides an orderly and predictable means to facilitate the evaluation of user-defined events. The *when* clause, provided by the Neuron C language, is used to specify such events. For more information on the operation of the scheduler, refer to the *Neuron C Programmer's Guide*.

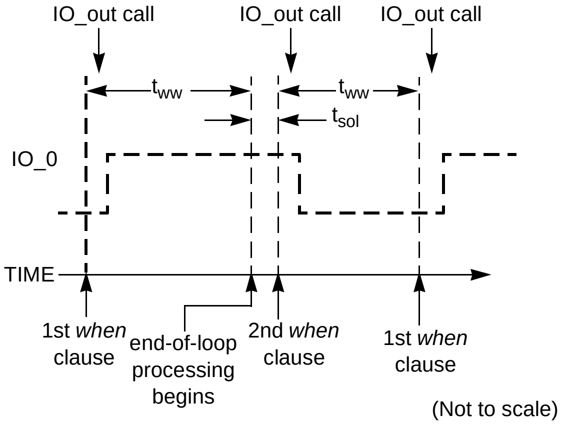
There is a finite latency associated with the operation of the scheduler. The time required for the scheduler to evaluate the same *when* clause in a particular user application code is, to a large extent, a function of the size of the user code, the total number of *when* clauses, and the state of the events associated with those *when* clauses. Therefore, it is impossible to specify a nominal value for this latency, as each application will have its own distinct behavior under different circumstances.

The best case latency can be viewed in several ways, each exposing a different aspect of the scheduler operation. A simple example consists of having an application program consisting of two *when* clauses, both of which always evaluate to TRUE, as shown below.

```
IO_0 output bit testbit;
when (TRUE) {
   io_out(testbit, 1);
}
```

```
when (TRUE) {
   io_out (testbit, 0);
}
```

Processing of *when* clauses is done in a round-robin fashion; therefore, the Neuron C code above performs alternating activation of the IO0 pin in order to isolate and extract the timing parameters associated with the scheduler. The waveform seen on pin IO0 of the Smart Transceiver, as a result of the above code, is shown in Figure 3.4.



Symbol	**Description**	**Typ @ 10MHz**
t_{ww}	*when*-clause to *when*-clause latency	940 µs
t_{sol}	Scheduler overhead latency (see text)	54 µs

Figure 3.4 *when*-Clause to *when*-Clause and Scheduler Overhead Latency

The *when*-clause to *when*-clause latency, t_{ww}, in this case includes the execution time of one `io_out()` function (65 µs latency at 10MHz) and is for an event that always evaluates to TRUE. The actual t_{ww} for a given application is driven by the actual task within the *when* statement as well as the *when* event which is evaluated.

The above example not only measures the best-case minimum latency between consecutive *when* clauses (whose events evaluate to TRUE), t_{ww}, but also reveals the scheduler's end-of-loop overhead latency, t_{sol}. As shown in Figure 3.4, t_{ww} is the off-time period of the output waveform and t_{sol} is the on-time of the output waveform, minus t_{ww}. This shows that the scheduler overhead latency, or the scheduler end-of-loop latency, occurs just before the execution of the last *when* clause in the program.

The latency associated with the return from the `io_out()` function is small, relative to that of the execution of the function call itself.

NOTE: Some I/O objects suspend application processing until the task is complete. This is because they are firmware-driven. These are bitshift, Neurowire, parallel, and serial I/O objects, I^2C, magcard, magtrack, Touch I/O, and Wiegand. They do not suspend network communication as this is handled by the network processor and the media access processor.

3.2.2 Firmware and Hardware Related I/O Timing Information

All I/O updates in the Smart Transceiver are performed by the Neuron firmware using system image function calls.

The total latency for a given function call, from start to end, can be broken down into two separate parts. The first is due to the processing time required before the actual hardware I/O update (read or write) occurs. The second delay is associated with the time required to finish the current function call and return to the application program.

Overall accuracy is always related to the accuracy of the Smart Transceiver’s CLK1 input. Timing diagrams are provided for all non-trivial cases to clarify the parameters given.

For more information on the operation of each of the I/O objects, refer to the *Neuron C Reference Guide*.

3.3 Direct I/O Objects

The timing numbers shown in this section are valid for both an explicit I/O call or an implicit I/O call through a *when* clause, and are assumed to be for a Smart Transceiver running at 10MHz.

3.3.1 Bit Input/Output

Pins IO0 – IO10 may be individually configured as single-bit input or output ports. Inputs may be used to sense TTL-level compatible logic signals from external logic, contact closures, and the like. Outputs may be used to drive external CMOS and TTL level compatible logic, switch transistors and very low current relays to actuate higher-current external devices such as stepper motors and lights. The high (20mA) current sink capability of pins IO0 – IO3 allows these pins to drive many I/O devices directly (refer to Figure 3.5). Figures 3.6 and 3.7 show the bit input and bit output latency times, respectively. These are the times from which *io_in()* or *io_out()* is called, until a value is returned. The direction of bit ports may be changed between input and output dynamically under application control.

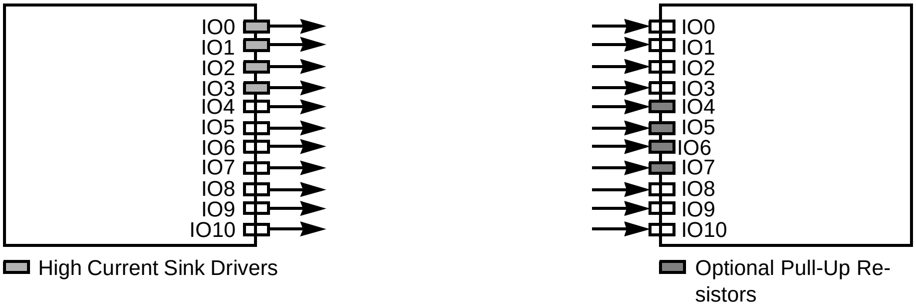
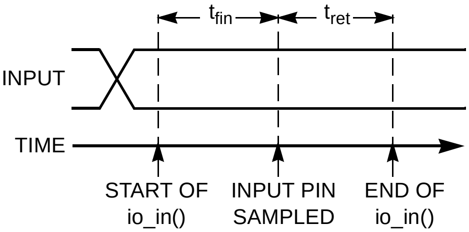


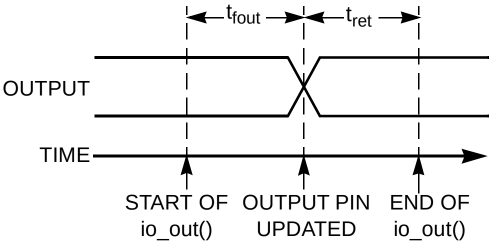
Figure 3.5 Bit I/O

WARNING: After a Reset, the Smart Transceiver performs a self-test which includes enabling the 104-107 pull-up resistors. This could cause a positive level change.



Symbol	Description	Typ @ 10MHz
t_{fin}	Function call to sample IO0 – IO10	 41 µs
t_{ret}	Return from function IO0 IO1 IO2 IO3 IO4 IO5 IO6 IO7 IO8 IO9 IO10	 19 µs 23.4 µs 27.9 µs 32.3 µs 36.7 µs 41.2 µs 45.6 µs 50 µs 19 µs 23.4 µs 27.9 µs

Figure 3.6 Bit Input Latency Values



Symbol	Description	Typ @ 10MHz
t_{fout}	Function call to update IO3 – IO5 All others	 69 µs 60 µs
t_{ret}	Return from function IO0 – IO10	 5 µs

Figure 3.7 Bit Output Latency Values

3.3.2 Byte Input/Output

Pins IO0 – IO7 may be configured as a byte-wide input or output port, which may be read or written using integers in the range 0 to 255. This is useful for driving devices that require ASCII data, or other data, eight bits at a time. For example, an alphanumeric display panel can use byte function for data, and use pins IO8 – IO10 in bit function for control and addressing. See Figures 3.8, 3.9, and 3.10. IO0 represents the LSB of data. The direction of a byte port may be changed between input and output dynamically under application control.

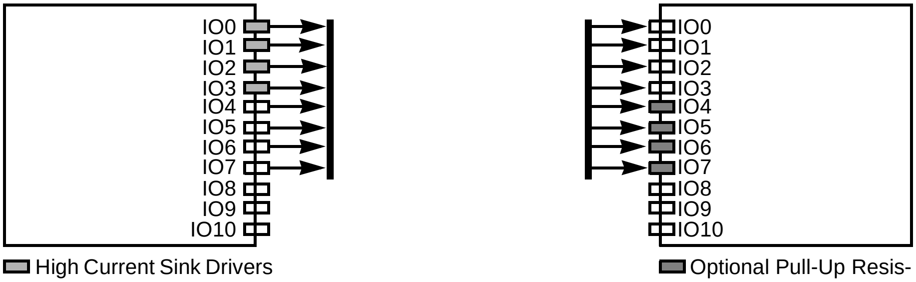
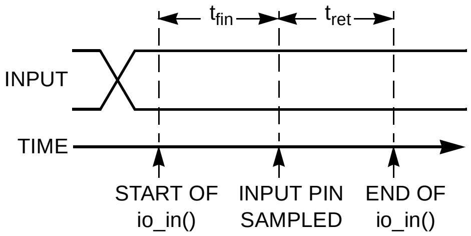
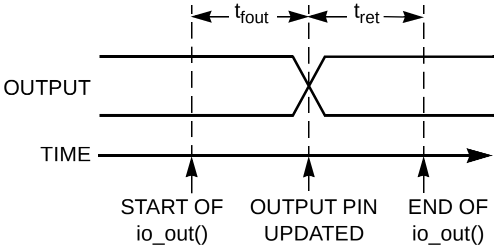


Figure 3.8 Byte I/O



Symbol	**Description**	**Typ @ 10MHz**
t_{fin}	Function call to input sample	24 µs
t_{ret}	Return from function	4 µs

Figure 3.9 Byte Input Latency Values

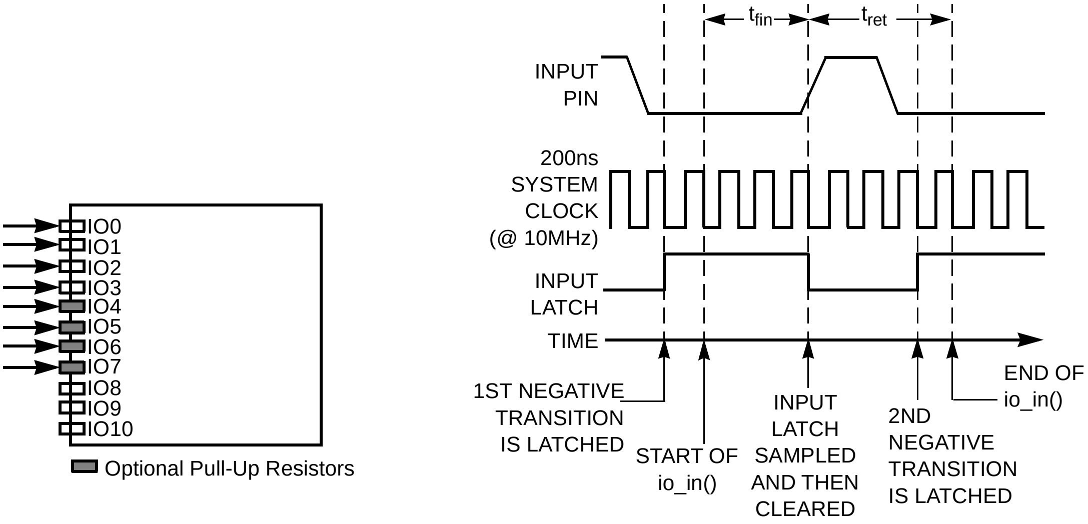


Symbol	Description	Typ @ 10MHz
t_{fout}	Function call to update	57 µs
t_{ret}	Return from function	5 µs

Figure 3.10 Byte Output Latency Values

3.3.3 Leveldetect Input

Pins IO0 – IO7 may be individually configured as leveldetect input pins, which latch a negative-going transition of the input level with a minimal low pulse width of 200ns, with a Smart Transceiver clocked at 10MHz. The application can therefore detect short pulses on the input which might be missed by software polling. This is useful for reading devices, such as proximity sensors. **This is the only direct I/O object which is latched before it is sampled.** The latch is cleared during the *when* statement sampling and can be set again immediately after, if another transition should occur. See Figure 3.11.


IO0
IO1
IO2
IO3
IO4
IO5
IO6
IO7
IO8
IO9
IO10
Optional Pull-Up Resistors
t_{fin}
t_{ret}
INPUT PIN
200ns SYSTEM CLOCK (@ 10MHz)
INPUT LATCH
TIME
1ST NEGATIVE TRANSITION IS LATCHED
START OF io_in()
INPUT LATCH SAMPLED AND THEN CLEARED
2ND NEGATIVE TRANSITION IS LATCHED
END OF io_in()

Symbol	**Description**	**Typ @ 10MHz**
t_{fin}	Function call to sample	
	IO0	35 µs
	IO1	39.4 µs
	IO2	43.9 µs
	IO3	48.3 µs
	IO4	52.7 µs
	IO5	57.2 µs
	IO6	61.6 µs
	IO7	66 µs
t_{ret}	Return from function	32 µs

Figure 3.11 Leveldetect Input Latency Values

3.3.4 Nibble Input/Output

Groups of four consecutive pins between IO0 – IO7 may be configured as nibble-wide input or output ports, which may be read or written to using integers in the range 0 to 15. This is useful for driving devices that require BCD data, or other data four bits at a time. For example, a 4x4 key switch matrix may be scanned by using one nibble to generate an output (row select — one of four rows), and one nibble to read the input from the columns of the switch matrix. See Figures 3.12, 3.13, and 3.14.

The direction of nibble ports may be changed between input and output dynamically under application control (see the *Neuron C Programmer's Guide*). The LSB of the input data is determined by the object declaration and can be any of the IO0 – IO4 pins.

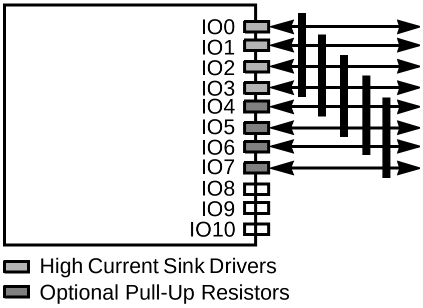
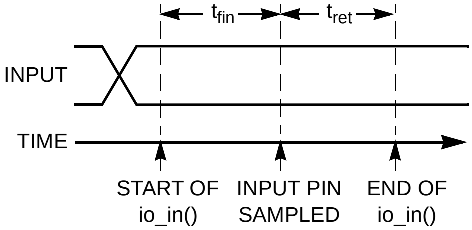
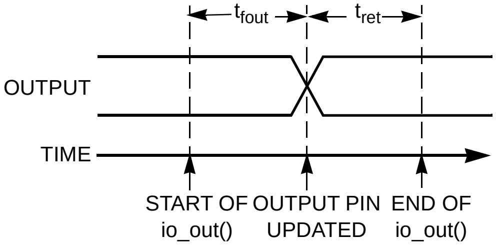


Figure 3.12 Nibble I/O



Symbol	**Description**	**Typ @ 10MHz**
t_{fin}	Function call to sample IO0 – IO4	41 µs
t_{ret}	Return from function	
	IO0	18 µs
	IO1	22.8 µs
	IO2	27.5 µs
	IO3	32.3 µs
	IO4	37 µs

Figure 3.13 Nibble Input Latency Values



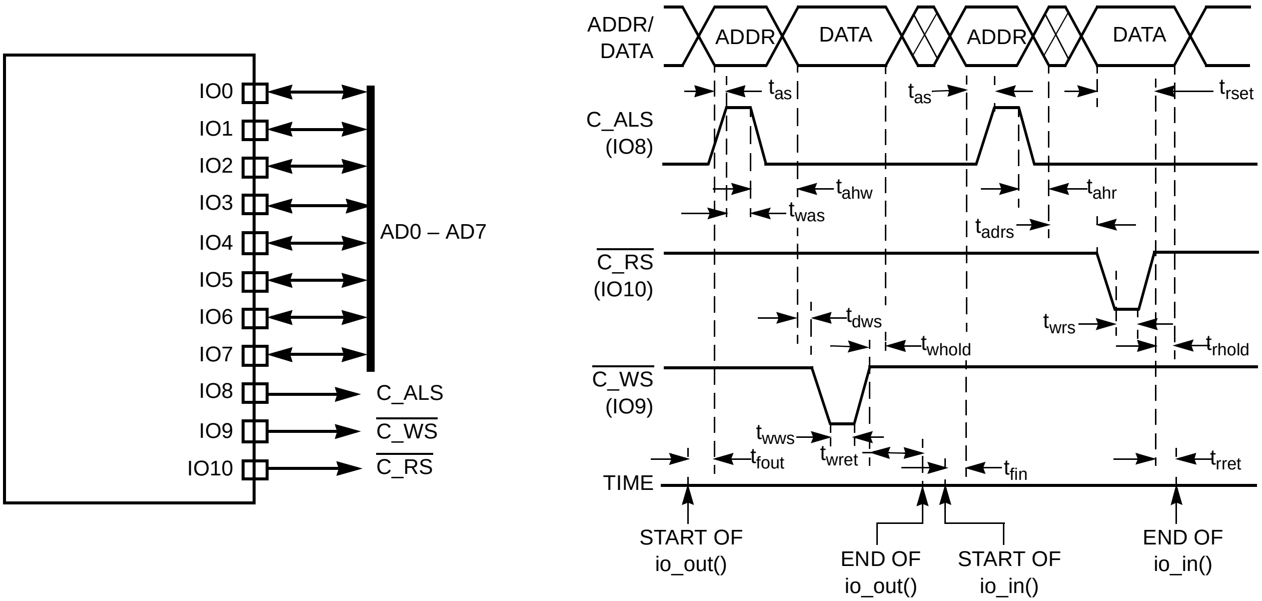
Symbol	Description	Typ @ 10MHz
t_{fout}	Function to update IO0 IO1 IO2 IO3 IO4	 78 µs 89.8 µs 101.5 µs 113.3 µs 125 µs
t_{ret}	Return from function IO0 – IO4	 5 µs

Figure 3.14 Nibble Output Latency Values

3.4 Parallel I/O Objects

3.4.1 Muxbus Input/Output

This I/O object provides a means of performing parallel I/O data transfers between the Smart Transceiver and an attached peripheral device or processor (see Figure 3.15). Unlike the parallel input/output object, which makes use of a token-passing scheme for ensuring synchronization, the muxbus input/output enables the Smart Transceiver to essentially be in control of all read and write operations at all times. This relieves the burden of protocol handling from the attached device and results in an easier-to-use interface at the expense of data throughput capacity. The data bus remains in the last state used.



NOTE: Data is latched 4.8 µs after the falling edge of $\overline{C\_RS}$.

Symbol	**Description**	**Min**	**Typ**	**Max**
t_{fout}	io_out() to valid address	—	26.4 µs	—
t_{as}	Address valid to address strobe	—	10.8 µs	—
t_{ahw}	Address hold for write	—	4.8 µs	—
t_{ahr}	Address hold for read	—	6.6 µs	—
t_{was}	Address strobe width	—	6.6 µs	—
t_{wrs}	Read strobe width	—	10.8 µs	—
t_{wws}	Write strobe width	—	10.8 µs	—
t_{dws}	Data valid to write strobe	—	6.6 µs	—
t_{rset}	Read setup time	10.8 µs	—	—
t_{whold}	Write hold time	4.2 µs	—	—
t_{rhold}	Read hold time	0 µs	—	—
t_{adrs}	Address disable to read strobe	—	7.2 µs	—
t_{fin}	io_in() to valid address	—	26.4 µs	—
t_{rret}	Function return from read	—	4.2 µs	—
t_{wret}	Function return from write	—	4.2 µs	—

Figure 3.15 Muxbus I/O Object

3.4.2 Parallel Input/Output

Pins IO0 – IO10 may be configured as a bidirectional 8-bit data and 3-bit control port for connecting to an external processor. The other processor may be a computer, microcontroller, or another Smart Transceiver (for gateway applications). The parallel interface may be configured in master, slave A, or slave B mode. Typically, two Smart Transceivers interface in master/slave A mode and a Smart Transceiver interfaces with another microprocessor in the slave B configuration, with the other microprocessor as the master. Handshaking is used in both modes to control the instruction execution, and application processing is suspended for the duration of the transfer (up to 255 bytes/transfer). Consult the *Neuron C Reference Guide* for detailed programming instructions.

Upon a reset condition, the master processor monitors the low transition of the handshake (HS) line from the slave, then passes a CMD_RESYNC (0x5A) for synchronization purposes. This must be done within 0.84 seconds after reset goes high with a Smart Transceiver slave running at 10MHz, to avoid a watchdog reset error condition (see the *Neuron C Programmer's Guide*). The CMD_RESYNC is followed by the slave acknowledging with a CMD_ACKSYNC (0x07). This synchronization ensures that both processors are properly reset before data transfer occurs. When interfacing two Smart Transceivers, these characters are passed automatically (refer to the flow table illustrated later in this section). However, when using parallel I/O to interface the Smart Transceiver to another microprocessor, that microprocessor must duplicate the interface signals and characters that are automatically generated by the Smart Transceiver's parallel I/O function.

For additional information, see the *Parallel I/O Interface to the Neuron Chip* engineering bulletin.

The timing numbers shown in this section are valid for both an explicit I/O call or an implicit I/O call through a *when* clause, and are assumed to be for a Smart Transceiver running at 10MHz.

3.4.2.1 Master/Slave A Mode

This mode is recommended when interfacing two Smart Transceivers. In a master/slave A configuration, the master drives IO8 as a chip select and IO9 to specify a read or write cycle, and the slave drives IO10 as a handshake (HS) acknowledgment (see Figure 3.16). The maximum data transfer rate is 1 byte per 4 processor instruction cycles, or 0.6 µs per byte at a 40MHz input clock rate. The data transfer rate scales proportionally to the input clock rate (a master write is a slave read). Timing for the case where the Smart Transceiver is the master (Figure 3.17), refers to measured output timing at 10MHz. After every byte write or byte read, the HS line is monitored by the master, to verify the slave has completed processing (when HS = 0) and the slave is ready for the next byte transfer. This is done automatically in Smart Transceiver-to-Smart Transceiver (master/slave A mode) data transfers. **The HS line should be pulled up (inactive) with a 10kΩ resistor to ensure proper resynch behavior after the slave resets.** Slave A timing is shown in Figure 3.18.

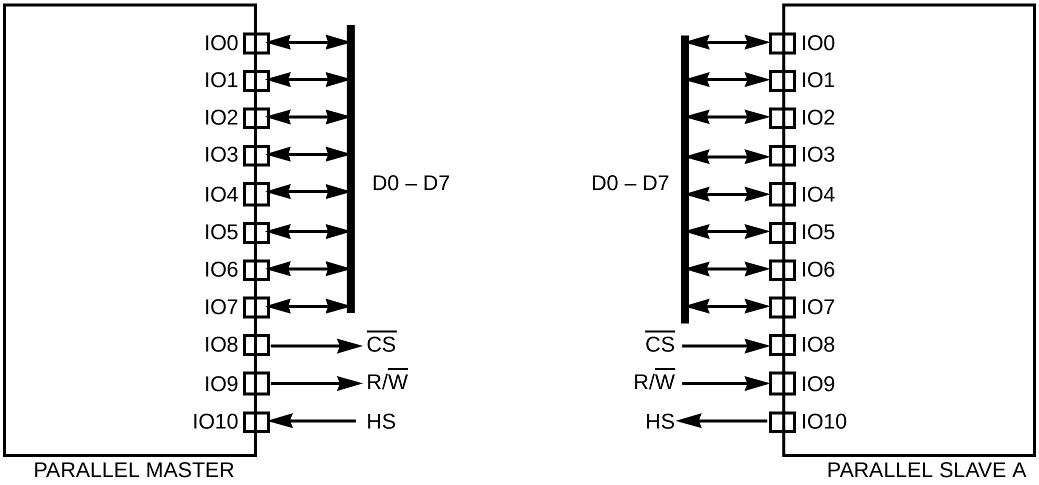
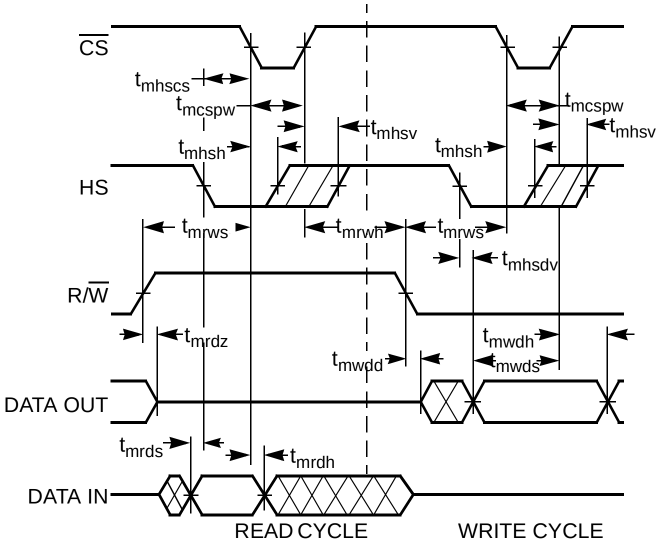


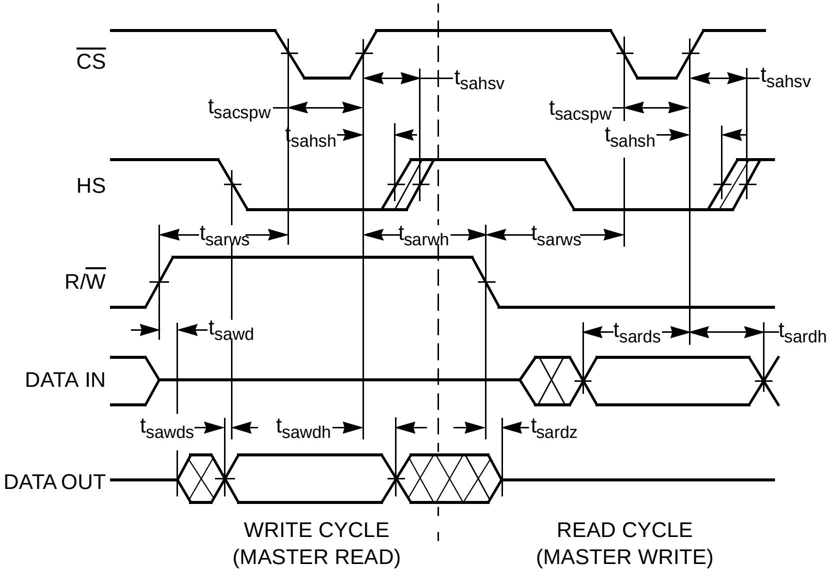
Figure 3.16 Parallel I/O — Master and Slave A



Symbol	Description	Min	Typ	Max
t_{mrws}	R/$\overline{W}$ setup before falling edge of $\overline{CS}$	150 ns	3 CLK1	—
t_{mrwh}	R/$\overline{W}$ hold after rising edge of $\overline{CS}$	100 ns	—	—
t_{mcspw}	$\overline{CS}$ pulse width	150 ns	2 CLK1	—
t_{mhsh}	HS hold after falling edge of $\overline{CS}$	0 ns	—	—
t_{mhsv}	HS checked by firmware after rising edge of $\overline{CS}$	150 ns	10 CLK1	—
t_{mrdz}	Master three-state DATA after rising edge of R/$\overline{W}$ (Notes 1, 2)	—	0	25 ns
t_{mrds}	Read data setup before falling edge of HS (Note 3)	0 ns	—	—
t_{mhscs}	HS low to falling edge of $\overline{CS}$ (Note 4)	2 CLK1	6 CLK1	—
t_{mrdh}	Read data hold after falling edge of $\overline{CS}$	0 ns	—	—
t_{mwdd}	Master drive of DATA after falling edge of R/$\overline{W}$ (Note 1)	150 ns	2 CLK1	—
t_{mhsdv}	HS low to data valid (Note 4)	—	50 ns	—
t_{mwds}	Write data setup before rising edge of $\overline{CS}$	150 ns	2 CLK1	—
t_{mwdh}	Write data hold after rising edge of $\overline{CS}$ (Note 6)	Note 1	—	—

Notes:
1. Refer to the *FT 3120 and FT 3150 Smart Transceiver Datasheet* for detailed measurement information.
2. For Smart Transceiver-to-Smart Transceiver operation, bus contention (t_{mrdz}, t_{sawdd}) is eliminated by firmware, ensuring that a zero state is present when the token is passed between the master and slave. See *Parallel I/O Interface to the Neuron Chip* engineering bulletin for further information.
3. HS high is used as a slave busy flag. If HS is held low, the maximum data transfer rate is 24 CLK1s (2.4 µs @ 10MHz) per byte. If HS is not used for a flag, caution should be taken to ensure the master does not initiate a data transfer before the slave is ready.
4. Parameters were added in order to aid interface design with the Smart Transceiver.
6. Master will hold output data valid during a write until the Slave device pulls HS low.
7. CLK1 represents the period of the Smart Transceiver input clock (100 ns at 10MHz).
8. In a master read, $\overline{CS}$ pulsing low acts like a handshake to flag the slave that data has been latched in.

Figure 3.17 Master Mode Timing



Symbol	Description	Min	Typ	Max
t_{sarws}	R/$\overline{W}$ setup before falling edge of $\overline{CS}$	25 ns	—	—
t_{sarwh}	R/$\overline{W}$ hold after rising edge of $\overline{CS}$	0 ns	—	—
t_{sacspw}	$\overline{CS}$ pulse width	45 ns	—	—
t_{sahsh}	HS hold after rising edge of $\overline{CS}$	0 ns	—	—
t_{sahsv}	HS valid after rising edge of $\overline{CS}$	—	—	50 ns
t_{sawdd}	Slave A drive of DATA after rising edge of R/$\overline{W}$ (Notes 1, 2)	0 ns	5 ns	—
t_{sawds}	Write data valid before falling edge of HS	150 ns	2 CLK1	—
t_{sawdh}	Write data valid after rising edge of $\overline{CS}$	150 ns (Note 3)	2 CLK1	—
t_{sardz}	Slave A three-state DATA after falling edge of R/$\overline{W}$ (Note 1)	—	—	50 ns
t_{sards}	Read data setup before rising edge of $\overline{CS}$	25 ns	—	—
t_{sardh}	Read data hold after rising edge of $\overline{CS}$	10 ns	—	—

Notes:

1. Refer to the *FT 3120 and FT 3150 Smart Transceiver Datasheet* for detailed measurement information.
2. For Smart Transceiver-to-Smart Transceiver operation, bus contention (t_{mrdz}, t_{sawdd}) is eliminated by firmware, ensuring that a zero state is present when the token is passed between the master and slave. See *Parallel I/O Interface to the Neuron Chip* engineering bulletin for further information.
3. If $t_{sarwh} < 150$ ns, then $t_{sawdh} = t_{sarwh}$.
5. CLK1 represents the period of the Smart Transceiver input clock (100 ns at 10MHz).
6. In slave A mode, the HS signal is high a minimum of 4 CLK1 periods. The typical time HS is high during consecutive data reads or consecutive data writes is also 4 CLK1 periods.

Figure 3.18 Slave A Mode Timing

The following is a pair of example programs that transfer data in a parallel I/O master/slave A configuration. The code is for two LonBuilder emulators hardwired as shown in Figure 3.16. The master program writes the test_data to the slave's input buffer (as the master owns the token after reset and has the first option to write on the bus) and the slave then outputs data to the master's input buffer. The buffers can be viewed through the LonBuilder debugger to verify the transfer was complete. The master transmits [5,1,1,1,1,1] to the slave and the slave transmits [7,1,2,3,4,5,6,7,0,0,0,0,0,0] to the master. The first byte indicates the number of bytes being passed; the following non-zero valued bytes in this example are the actual data transferred. The remaining length of the array, if any, is filled with 0s. The master program writes once to the slave and reads once from the slave. To implement continuous writes and reads, add an *io_out_request()* function call after the *io_in()* function call in the master program.

```
/* This is the master program. After reset, the buffer is filled with 1s
and then the buffer is written to the slave. The master then reads the
slave’s buffer. The master’s output buffer should contain [5,1,1,1,1,1];
the input buffer should contain [7,1,2,3,4,5,6,7,0,0,0,0,0,0].
*/
IO_0 parallel master parallel_bus;
#define TEST_DATA 1     // data to be written in output buffer
#define MAX_IN 13       // maximum length of input data expected
#define OUT_LEN 5       // output length can be equal to or less than the
                        // max
#define MAX_OUT 5       // maximum array length
struct parallel_out     // output structure
{
   unsigned int len;                // actual length of data to be output
   unsigned int buffer[MAX_OUT];    //  array setup for max length of
                                    // data to be output
}p_out;                             // output structure name
struct parallel_in                  // input structure
{
   unsigned int len;                // actual buffer length to be input
   unsigned int buffer[MAX_IN];     // maximum input array

}p_in;                              // input structure name

unsigned int i;

when (reset)
{
   p_out.len=OUT_LEN;               // assign output length
   for(i=0; i<OUT_LEN; ++i)         // fill output buffer with 1s
   p_out.buffer[i]=TEST_DATA;
   io_out_request(parallel_bus);     // request to output buffer
}

when (io_out_ready(parallel_bus))
{
   io_out(parallel_bus, &p_out);     // output buffer when slave is ready
}

when (io_in_ready(parallel_bus))
{
   p_in.len=MAX_IN;                  // declare the maximum input
                                     // buffer acceptable
   io_in(parallel_bus, &p_in);       // store input data in buffer
}                                    // end of program

/* This is the slave program. After reset, the output buffer is filled
with data and then the slave reads from the master. The slave then writes
to the master. The slave’s input buffer should contain [5,1,1,1,1,1]; the
output buffer should contain [7,1,2,3,4,5,6,7,0,0,0,0,0,0].
*/
IO_0 parallel slave parallel_bus;
#define MAX_IN 5       // maximum length of input data expected
#define OUT_LEN 7      // output length can be equal to or less than the
                       // max
#define MAX_OUT 13     // maximum array length
struct parallel_out    // output structure
{
   unsigned int len;   // actual length of data to be output
   unsigned int buffer[MAX_OUT];     // array setup for max length of data
                                     // to be output

}p_out;                              // output structure name

struct parallel_in                   // input structure
{
   unsigned int len;                 // actual length of buffer to be
                                     // input
   unsigned int buffer[MAX_IN];      // maximum input array

}p_in;                               // input structure name

unsigned int i;
when (reset)
{
   p_out.len=OUT_LEN;                // assign output length
   for(i=0; i<OUT_LEN; ++i)          // fill output buffer with 1s
   p_out.buffer[i]=i+1;
}

when (io_out_ready(parallel_bus))
{
   io_out(parallel_bus, &p_out);     // output buffer
}

when (io_in_ready(parallel_bus))
{
   p_in.len=MAX_IN;                  // declare the maximum input buffer
                                     // acceptable
   io_in(parallel_bus, &p_in);       // store input data in buffer
   io_out_request(parallel_bus);     // request to output buffer
}                                    // end of program
```

Debugging the Above Programs: If a watchdog timeout occurs on either LonBuilder emulator, simultaneously reset the two emulators using the reset pushbutton switches on the face of the emulators. Both JP1 and JP2 on the emulator boards should be disconnected for this application.

3.4.2.2 Slave B Mode

The slave B mode is recommended for interfacing a Smart Transceiver acting as the slave to another microprocessor acting as the master. When configured in slave B mode, the Smart Transceiver accepts IO8 as a chip select and IO9 to specify whether the master will read or write, and accepts IO10 as a register select input. When $\overline{CS}$ is asserted and either IO10 is low or IO10 is high and R/$\overline{W}$ is low, pins IO0 – IO7 form the bidirectional data bus. When IO10 is high, R/$\overline{W}$ is high, and $\overline{CS}$ is asserted, IO0 is driven as the HS acknowledgment signal to the master.

The Smart Transceiver may appear as two registers in the master’s address space; one of the registers being the read/write data register, and the other being the read-only status register. Therefore, reads by the master to an odd address access the status register for handshaking acknowledgments and all other reads or writes access the data register for I/O transfers. The LSB of the control register, which is read through pin IO0, is the HS bit. The master reads the HS bit after every master read or write. **The D0/HS line should be pulled up (inactive) with a 10kΩ resistor to ensure proper resynch behavior after resets.**

When acting as a slave to a different microprocessor, the Smart Transceiver slave B mode handles all handshaking and token passing automatically. However, the master microprocessor must read the HS bit after each transaction and must also internally track the token passing. This mode is designed for use with a master processor that uses memory-mapped I/O, as the LSB of the master's address bus is typically connected to the Smart Transceiver's IO10 pin. This is illustrated in Figures 3.19 and 3.20.

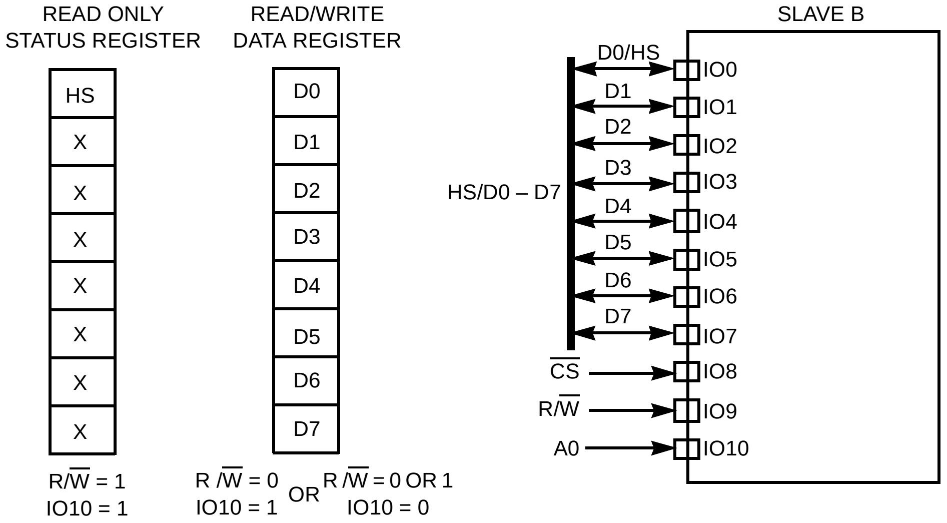


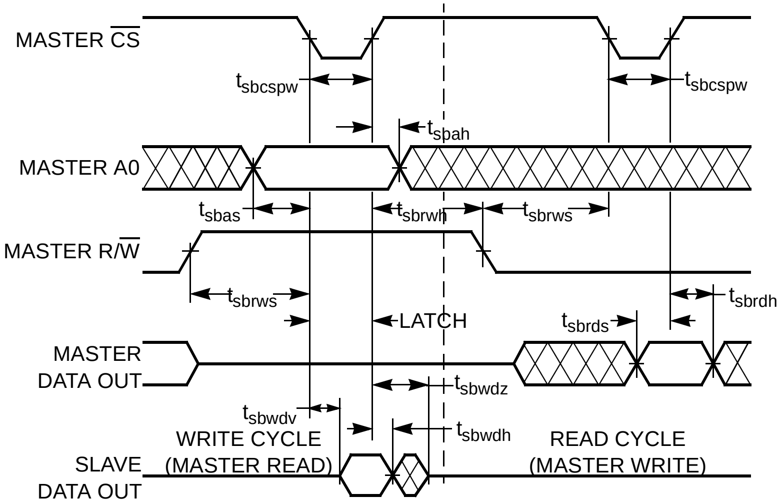
Figure 3.19 Parallel I/O Master/Slave B (FT 3120/FT 3150 Smart Transceiver as Memory-Mapped I/O Device)

3.4.2.3 Token Passing

Virtual token passing is implemented to eliminate the possibility of data bus contention. The token is owned by the master after synchronization and is passed between the master and slave devices. After each data transfer is completed, the token owner writes an end of message (EOM) (0x00) to indicate that the transfer is complete. The EOM is never read. Instead, "processing the EOM" indicates passing of the token. Token passing can be achieved by executing either a data packet or a NULL transfer. Only the owner of the token can write to the bus. Therefore, when the master performs two writes of data (1 – 255 bytes each) a dummy read cycle (NULL character = 0x00) must be inserted between them in order to pass the token. Token passing is executed automatically in a Smart Transceiver-to-Smart Transceiver interface. Refer to section 3.4.2.5, *Data Transferring*, for master/slave flow transactions.

3.4.2.4 Handshaking

Handshaking allows the master to monitor the slave between every byte transfer, ensuring that both processors are ready for the byte to be transferred. If the master owns the token, the master waits for the HS from the slave before writing data to the bus. If the slave owns the token, the master monitors the low transition of the HS before reading the bus. In master or slave A mode, the Smart Transceiver HS line is pin IO10. In slave B mode, the Smart Transceiver HS bit is monitored on IO0 which corresponds to the least significant data bit of the status register.



Symbol	Description	Min	Typ	Max
t_{sbrws}	R/$\overline{\text{W}}$ setup before falling edge of $\overline{\text{CS}}$ FT 3150 and FT 3120 Smart Transceivers	0 ns	—	—
t_{sbrwh}	R/$\overline{\text{W}}$ hold after rising edge of $\overline{\text{CS}}$	0 ns	—	—
t_{sbcspw}	$\overline{\text{CS}}$ pulse width	Note 1	—	—
t_{sbas}	A0 setup to falling edge of $\overline{\text{CS}}$	10 ns	—	—
t_{sbah}	A0 hold after rising edge of $\overline{\text{CS}}$	0 ns	—	—
t_{sbwdv}	$\overline{\text{CS}}$ to write data valid	—	—	50 ns
t_{sbwdh}	Write data hold after rising edge of $\overline{\text{CS}}$ (Notes 2, 3)	0 ns	30 ns	—
t_{sbwdz}	$\overline{\text{CS}}$ rising edge to Slave B release data bus (Note 2)	—	—	50 ns
t_{sbrds}	Read data setup before rising edge of $\overline{\text{CS}}$	25 ns	—	—
t_{sbrdh}	Read data hold after rising edge of $\overline{\text{CS}}$	10 ns	—	—

Notes:

1. The slave B write cycle (master read) $\overline{\text{CS}}$ pulse width is directly related to the slave B write data valid parameter and master read setup parameter. To calculate the write cycle $\overline{\text{CS}}$ duration needed for a special application use:
   $t_{sbcspw} = t_{sbwdv}$ + master's read data setup before rising edge of $\overline{\text{CS}}$
   Refer to the master's specification data book for the master read setup parameter. The slave read cycle minimum $\overline{\text{CS}}$ pulse width = 50 ns.
2. Refer to the *FT 3120 and FT 3150 Smart Transceiver Datasheet* for detailed measurement information.
3. The data hold parameter, t_{sbwdh}, is measured to the disable levels shown in the *FT 3120 and FT 3150 Smart Transceiver Datasheet*, rather than to the traditional data invalid levels.
4. In a slave B write cycle the timing parameters are the same for a control register (HS) write as for a data write.
5. Special applications: Both the state of $\overline{\text{CS}}$ and R/$\overline{\text{W}}$ determine a slave B write cycle. If $\overline{\text{CS}}$ can not be used for a data transfer, then toggling the R/$\overline{\text{W}}$ line can be used with no changes to the hardware. In other words, if $\overline{\text{CS}}$ is held low during a slave B write cycle, a positive pulse (low to high to low) on R/$\overline{\text{W}}$ can execute a data transfer. The low to high transition on R/$\overline{\text{W}}$ causes slave B to drive data with the same timing parameters as t_{sbwdv} (redefined R/$\overline{\text{W}}$ to write data valid). Likewise, the falling edge of R/$\overline{\text{W}}$ causes slave B to release the data bus with the same timing limits as the $\overline{\text{CS}}$ rising edge in t_{sbwdz}. This scenario is only true for a slave B write cycle and is not applicable to a slave B read cycle or any slave A data transitions. This application may be helpful if the master has separate read and write signals but no $\overline{\text{CS}}$ signal. Caution must be taken to ensure the bus is free before transfers to avoid bus contention.

Figure 3.20 Slave B Mode Timing

3.4.2.5 Data Transferring

The data transfer operation between the master and the slave is accomplished through the use of a virtual write token-passing protocol. The write token is passed alternatively between the master and the slave on the bus in an infinite ping-pong fashion. The owner of the token has the option of writing a series of data bytes, or alternatively, passing the write token without any data. Figure 3.21 illustrates the sequence of operations for this token passing protocol.

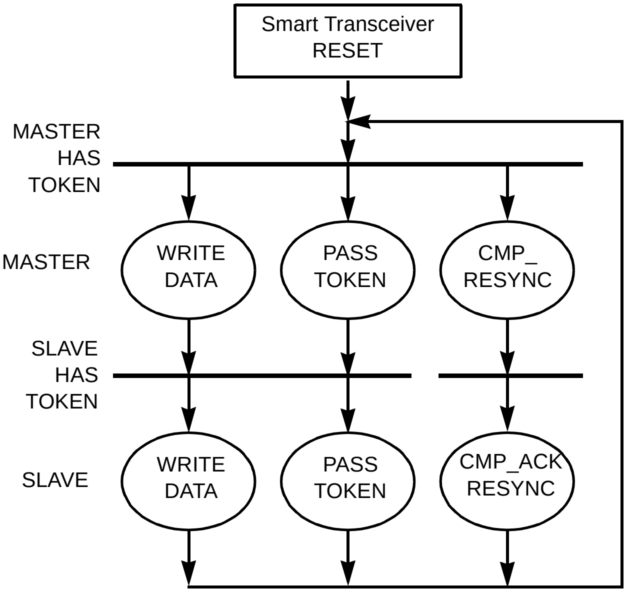


Figure 3.21 Handshake Protocol Sequence Between the Master and the Slave

Once in possession of the write token, the device (Smart Transceiver or a host processor) can transfer up to 255 bytes of data. The stream of data bytes is preceded by the command and length bytes. The token holder keeps possession of the token until all data bytes have been written, after which the token is passed to the attached device. The same process may now be repeated by the other side or, alternatively, the token can be passed back without any data. The timing relationship between the various Smart Transceiver signals involved in this process is shown in the following timing diagrams.

Resynchronization Procedure: The following procedure applies to master/slave A and master/slave B configuration. The master initiates the resynchronization with a RESYNC

(0x5A) command, and the slave acknowledges with an ACKSYNC (0x07). If the slave does not respond, the master continues to send the RESYNC until the slave responds correctly.

MASTER	**SLAVE**	
(Owns Token)		
Write RESYNC		// master initiates resynchronization (0x5A)
	Read RESYNC	
Write EOM		// end of message (EOM=0x00)
	Process EOM	
	Write ACKSYNC	// slave acknowledges resynching (0x07)
Read ACKSYNC		
	Write EOM	
Process EOM		// master owns token when reset
(Owns Token)		

Master writes buffer to slave: Enter RD/_WR=0.

MASTER	**SLAVE**	
(Owns Token)		
Write XFER		// master has data to write (XFER=0x01)
	Read XFER	
Write (length)		// length=number of bytes of data
	Read (length)	
Write (data_0)		// master begins data transfer to slave
	Read (data_0)	
.	.	
.	.	
.	.	
Write (data_n)		// last byte of data to be transferred
	Read (data_n)	
Write EOM		// end of data transfer (EOM=0x00)
	Process EOM// exchange token	
	(Owns Token)	

Slave writes buffer to master: Enter RD/_WR=1.

MASTER	**SLAVE**	
	(Owns Token)	
	Write XFER	// slave has data to write (XFER=0x01)

Read XFER		
	Write (length)	// length=number of bytes of data
Read (length)		
	Write (data_0)	// slave begins writing data to master
Read (data_0)		
.	.	
.	.	
.	.	
	Write (data_n)	// last byte of data to be transferred
Read (data_n)		
	Write EOM	// end of data transfer
Process EOM		// exchange token
(Owns Token)		

Master passes token to slave: Entry same as when master writes buffer to slave.

MASTER	**SLAVE**	
(Owns Token)		
Write NULL		// master has no data to send to slave
	Read NULL	// NULL=0x00
Write EOM		// end of message (EOM=0x00)
	Process EOM	// exchange token
	(Owns Token)	

Slave passes token to master: Entry same as when slave writes buffer master.

MASTER	**SLAVE**	
	(Owns Token)	
	Write NULL	// slave has no data to send to the master
Read NULL		// NULL=0x00
	Write EOM	// end of message (EOM=0x00)
Process EOM		// exchange token
(Owns Token)		

3.5 Serial I/O Objects

The timing numbers shown in this section are valid for both an explicit I/O call or an implicit I/O call through a *when* clause, and are assumed to be for a Smart Transceiver running at 10MHz.

3.5.1 Bitshift Input/Output

Pairs of adjacent pins may be configured as serial input or output lines. The first pin of the pair can be IO0-IO6, IO8, or IO9, and is used for the clock (driven by the Smart Transceiver). The adjacent higher-numbered I/O pin is then used for up to 16 bits of serial data. The data rate may be configured as 1kbps, 10kbps, or 15kbps at a 10MHz input clock rate. The data rate scales proportionally to the input clock rate, for example: 4kbps, 40kbps, or 60kbps at a 40MHz input clock rate. The active clock edge may be specified as either rising or falling. This object is useful for transferring data to external logic employing shift registers. This function suspends application processing until the operation is complete. See Figures 3.22, 3.23, and 3.24.

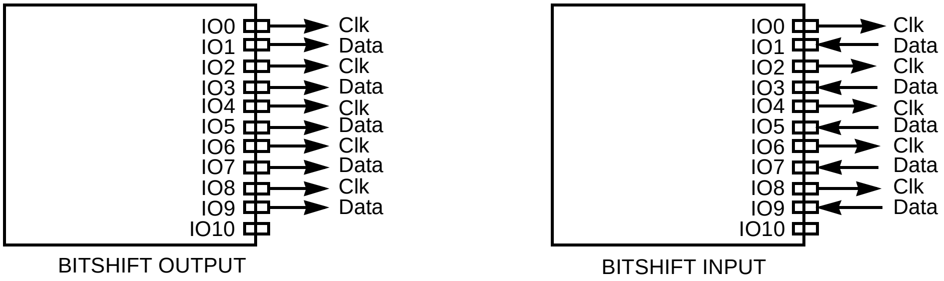
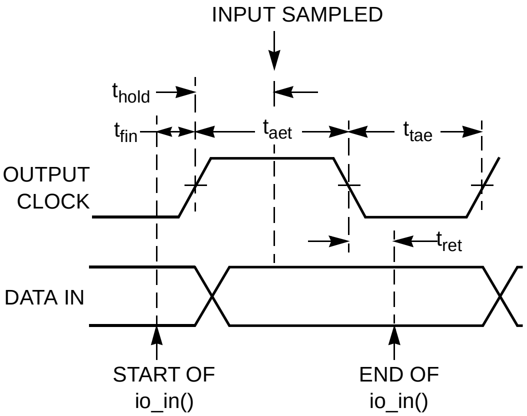


Figure 3.22 Bitshift I/O Examples

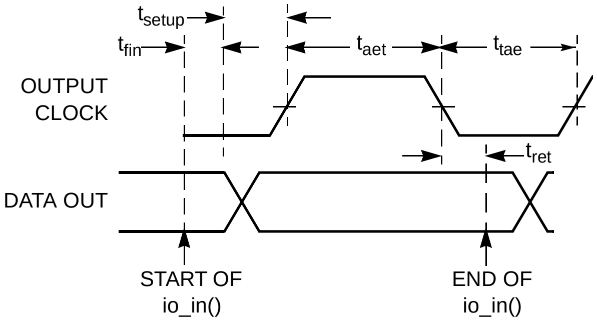
For bitshift input, the clock output is deasserted (to the inactive level) at the same time as the start of the first bit of data. For bitshift output, the clock output is initially inactive prior to the first bit of data (unless overridden by a bit output overlay).



Active clock edge assumed to be positive in the above diagram

Symbol	Description	Typ @ 10MHz
t_{fin}	Function call to first edge	156.6 µs
t_{ret}	Return from function	5.4 µs
t_{hold}	Active clock edge to sampling of input data 15 kbps bit rate 10 kbps bit rate 1 kbps bit rate	 9 µs 40.8 µs 938.2 µs
t_{aet}	Active clock edge to next clock transition 15 kbps bit rate 10 kbps bit rate 1 kbps bit rate	 31.8 µs 63.6 µs 961 µs
t_{tae}	Clock transition to next active clock edge 15 kbps bit rate 10 kbps bit rate 1 kbps bit rate	 14.4 µs 14.4 µs 14.4 µs
f	Clock frequency = $1/(t_{aet} + t_{tae})$ 15 kbps bit rate 10 kbps bit rate 1 kbps bit rate	 21.6 kHz 12.8 kHz 1.03 kHz

Figure 3.23 Bitshift Input Latency Values



Active clock edge assumed to be positive in the above diagram

Symbol	**Description**	**Typ @ 10MHz**
t_{fin}	Function call to first data out stable 16-bit shift count 1-bit shift count	 185.3 µs 337.6 µs
t_{ret}	Return from function	10.8 µs
t_{setup}	Data out stable to active clock edge 15 kbps bit rate 10 kbps bit rate 1 kbps bit rate	 10.8 µs 10.8 µs 10.8 µs
t_{aet}	Active clock edge to next clock transition 15 kbps bit rate 10 kbps bit rate 1 kbps bit rate	 10.2 µs 42 µs 939.5 µs
t_{tae}	Clock transition to next active clock edge 15 kbps bit rate 10 kbps bit rate 1 kbps bit rate	 34.8 µs 34.8 µs 34.8 µs
f	Clock frequency = $1/(t_{aet} + t_{tae})$ 15 kbps bit rate 10 kbps bit rate 1 kbps bit rate	 22 kHz 13 kHz 1.02 kHz

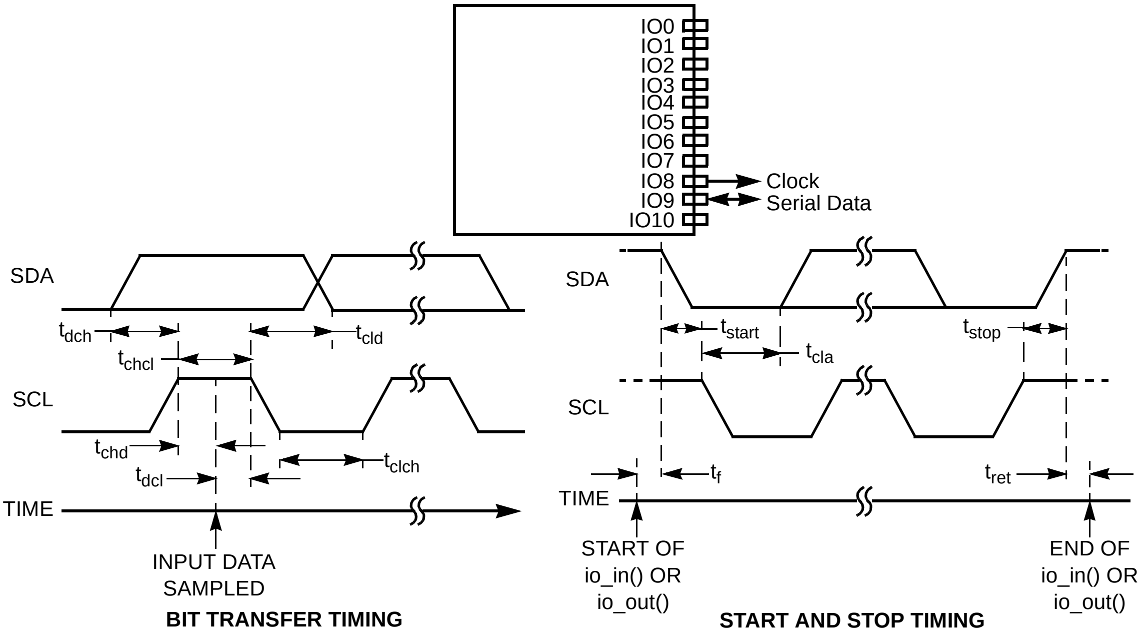
Figure 3.24 Bitshift Output Latency Values

3.5.2 I^{2}C Input/Output

This I/O object is used to interface the Smart Transceiver to any device which adheres to Philips Semiconductor's Inter-Integrated Circuit (I^{2}C) bus protocol. The Smart Transceiver is always the master, with IO8 being the serial clock (SCL) and IO9 the serial data (SDA). These I/O lines are operated in the open-drain mode in order to accommodate the special requirements of the I^{2}C protocol. With the exception of two pull-up resistors, no additional external components are necessary for interfacing the Smart Transceiver to an I^{2}C device.

Up to 255 bytes of data may be transferred at a time. At the start of all transfers, a right-justified 7-bit I^{2}C address argument is sent out on the bus immediately after the I^{2}C "start condition."

For more information on this protocol, refer to Philips Semiconductor's I^{2}C documentation.



Parameter	Description	Min	Typ	Max

t_f	I/O call to start condition
	io_in()
	io_out()
		
		—
		—
			
			54.6 µs
			43.4 µs
				
				—
				—

t_{start}	End of start condition
	io_in()
	io_out()
		
		5.4 µs
		5.4 µs
			
			—
			—
				
				—
				—

t_{cla}	End of start to start of address
	io_in()
	io_out()
		
		24.0 µs
		24.0 µs
			
			—
			—
				
				—
				—

t_{cld}	SCL low to data for io_out()	24.6 µs	—	—
t_{dch}	Data to SCL high for io_out()	7.2 µs	—	—
t_{chcl}	Clock high to clock low for io_out()	12.6 µs	—	—
t_{chd}	SCL high to data sampling for io_in()	13.2 µs	—	—
t_{dcl}	Data sample to SCL low for io_in()	7.2 µs	—	—
t_{clch}	Clock low to clock high for io_in()	24.0 µs	—	—

t_{stop}	Clock high to data
	io_in()
	io_out()
		
		12.6 µs
		12.6 µs
			
			—
			—
				
				—
				—

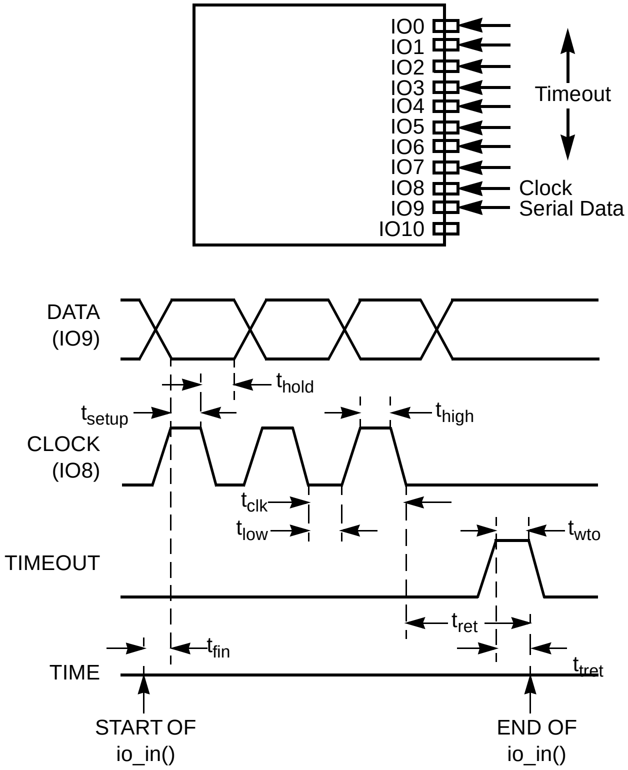
t_{ret}	SDA high to return from function
	io_in()
	io_out()
		
		—
		—
			
			—
			—
				
				4.2 µs
				4.2 µs

Figure 3.25 I^{2}C I/O Object

3.5.3 Magcard Input

This I/O object is used to transfer synchronous serial data from an ISO 7811 Track 2 magnetic stripe card reader in real time. The data is presented as a data signal input on pin IO9, and a clock, or a data strobe, signal input on pin IO8. The data on pin IO9 is clocked on or just following the falling (negative) edge of the clock signal on IO8, with the LSB first. In addition, any one of the pins IO0 – IO7 may be used as a timeout pin to prevent lockup in case of abnormal abort of the input bit stream during the input process.

Up to 40 characters may be read at one time. Both the parity and the Longitudinal Redundancy Check (LRC) are checked by the Smart Transceiver.



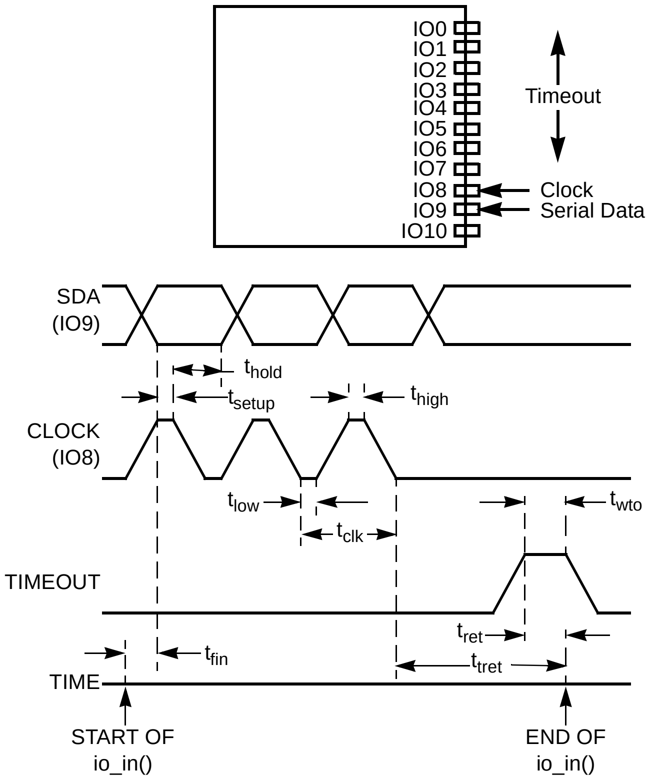
Symbol	Description	Min	Typ	Max
t_{fin}	Function call to first clock input	—	45.0 µs	—
t_{hold}	Data hold	0 µs	—	—
t_{setup}	Data setup	0 µs	—	—
t_{low}	Clock low width	60 µs	—	—
t_{high}	Clock high width	60 µs	—	—
t_{wto}	Width of timeout pulse	60 µs	—	—
t_{clk}	Clock period	120 µs	—	—
t_{tret}	Return from timeout	21.6 µs	—	81.6 µs
t_{ret}	Return from function	—	—	301.8 µs

Figure 3.26 Magcard Input Object

A Smart Transceiver operating at 10MHz can process a bit rate at up to 8334 bits/second (of a bit density of 75 bits/inch). This equates to a card velocity of 111 inches/second. Most magnetic card stripes contain a 15-bit sequence of zero data at the start of the card, allowing time for the application to start the card reading function. At 8334 bits/second, this period is about 1.8 ms. If the scheduler latency is greater than the 1.8 ms value, the `io_in( )` function will miss the front end of the data stream. The bit rate processing capability scales with input clock rate. For example: the bit rate may be up to 33,336 bps at 40MHz.

3.5.4 Magtrack1 Input

This input object type is used to read synchronous serial data from an ISO3554 magnetic stripe card reader. The data input is on pin IO9, and the clock, or data strobe, is presented as input on pin IO8. The data on pin IO9 is clocked in just following the falling edge of the clock signal on IO7, with the LSB first.



Symbol	Description	Min	Typ	Max
t_{fin}	Function call to first clock input	—	45.0 µs	—
t_{hold}	Data hold	t_{low}	—	t_{clk}
t_{setup}	Data setup	0 µs	—	—
t_{low}	Clock low width	31 µs	—	—
t_{high}	Clock high width	31 µs	—	—
t_{wto}	Width of timeout pulse	60 µs	—	—
t_{clk}	Clock period	138 µs	—	—
t_{tret}	Return from timeout	21.6 µs	—	81.6 µs
t_{ret}	Return from function	—	—	301.8 µs

Figure 3.27 Magtrack1 Input Object

The minimum period for the entire bit cycle (t_{clk}) is greater than the sum of t_{low} and t_{high}. The t_{setup} and t_{hold} times should be such that the data is stable for the duration of t_{low}.

Data are recognized in the IATA format as a series of 6-bit characters plus an even parity bit per character. The process begins when the start sentinel (hex 05) is recognized, and continues until the end sentinel (0x0F) is recognized. No more than 79 characters, including the 2 sentinels and the LRC character, will be read. The data is stored as right-justified bytes in the buffer space pointed to by the buffer pointer argument in the `io_in()` function with the parity stripped, and includes the start and end sentinels. This buffer should be 78 bytes long.

The magtrack1 input object optionally uses one of the I/O pins IO0 – IO7 as a timeout/abort pin. Use of this feature is suggested since the `io_in()` function will update the watchdog timer during clock wait states, and could result in a lockup if the card were to stop moving in the middle of the transfer process. If a logic 1 level is detected on the I/O timeout pin, the `io_in()` function will abort. This input can be a oneshot timer counter output, an R/C circuit, or a $\overline{\text{DATA\_VALID}}$ signal from the card reader.

A Smart Transceiver with a clock rate of 10MHz can process an incoming bit rate of up to 7246 bits/second when the strobe signal has a 1/3 duty cycle (t_{high} = 46 µs, t_{low} = 92 µs). At a bit density of 210 bits/inch, this translates to a card speed of 34.5 inches/second. The bit rate processing capability scales with Smart Transceiver input clock rate, for example: bit rate up to 28,984bps at 40MHz.

3.5.5 Neurowire (SPI Interface) Input/Output Object

The Neurowire object implements a full-duplex synchronous transfer of data to some peripheral device. It can operate as the master (drive a clock out) or as the slave (accept a clock in). In both master and slave modes, up to 255 bits of data may be transferred at a time. The Neurowire I/O suspends application processing until the operation is completed. The Neurowire object is useful for external devices, such as A/D, D/A converters, and display drivers incorporating serial interfaces that conform with Motorola's SPI or National Semiconductor Microwave^{TM} interface. See Figure 3.28.

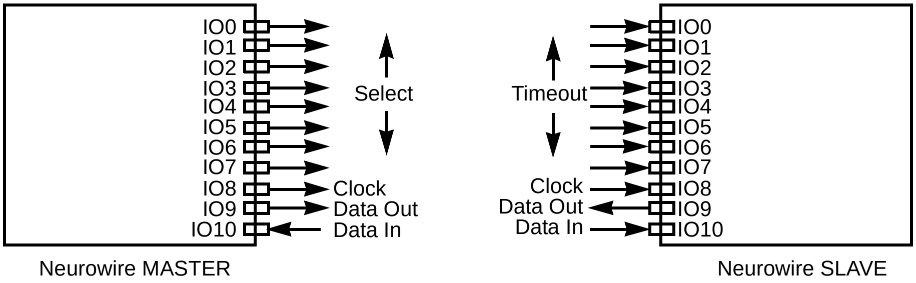
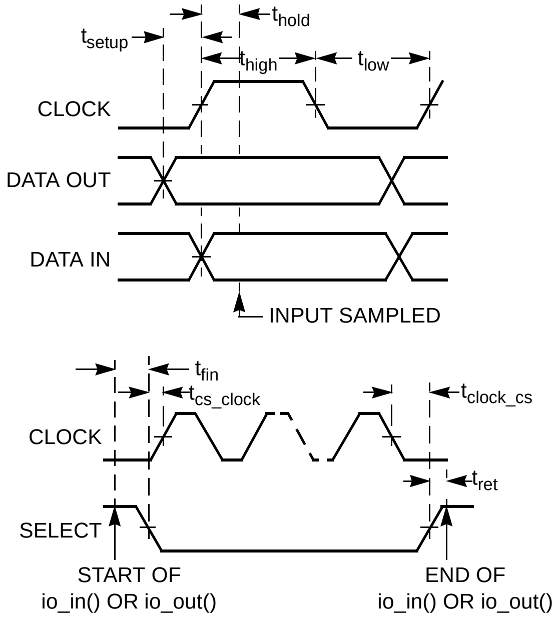


Figure 3.28 Neurowire I/O

3.5.5.1 Neurowire Master Mode

In Neurowire master mode, pin IO8 is the clock (driven by the Smart Transceiver), IO9 is the serial data output, and IO10 is the serial data input. Serial data is clocked out on pin IO9 at the same time as data is clocked in from pin IO10. Data is clocked by the rising edge of the clock signal by default. The *clockedge(-)* keyword changes the active edge of the clock to negative. In addition, one or more of the IO0 – IO7 pins may be used as a chip select, allowing multiple Neurowire devices to be connected on a three-wire bus. The clock rate may be specified as 1kbps, 10kbps, or 20kbps at an input clock rate of 10MHz; these scale proportionally with input clock, for example: 4kbps, 40kbps, or 80kbps at an input clock rate of 40MHz. See Figure 3.29.

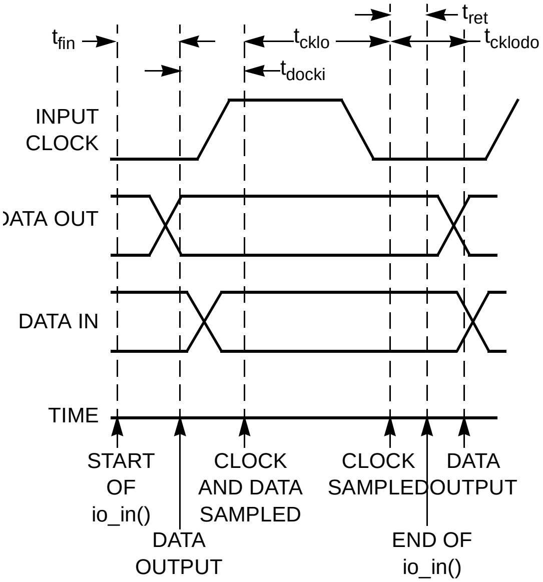


Parameter	Description	Typ
t_{fin}	Function call to $\overline{\text{CS}}$ active	69.9 µs
t_{ret}	Return from function	7.2 µs
t_{hold}	Active clock edge to sampling of input data 20 kbps bit rate 10 kbps bit rate 1 kbps bit rate	 11.4 µs 53.4 µs 960.6 µs
t_{high}	Period, clock high (active clock edge = 1) 20 kbps bit rate 10 kbps bit rate 1 kbps bit rate	 25.8 µs 67.8 µs 975.0 µs
t_{low}	Period, clock low (active clock edge = 1)	33.0 µs
t_{setup}	Data output stable to active clock edge	5.4 µs
t_{cs_clock}	Select active to first active clock edge	91.2 µs
t_{clock_cs}	Last clock transition to select inactive	81.6 µs
f	Clock frequency = $1/(t_{high} + t_{low})$ 20 kbps bit rate 10 kbps bit rate 1 kbps bit rate	 17.0 kHz 9.92 kHz 992 Hz

Figure 3.29 Neurowire (SPI) Master Timing

3.5.5.2 Neurowire Slave Mode

In Neurowire slave mode, pin IO8 is the clock (driven by the external master), IO9 is the serial data output, and IO10 is the serial data input. Serial data is clocked out on pin IO9 at the same time as data is clocked in from pin IO10. Data is clocked by the rising edge of the clock signal (default), which may be up to 18kbps at 10MHz. This data rate scales with Smart Transceiver input clock rate, for example: 72kbps at 40MHz. The `clockedge(-)` keyword changes the active edge of the clock to negative. One of the IO0 – IO7 pins may be designated as a timeout pin. A logic 1 level on the timeout pin causes the Neurowire slave I/O operation to be terminated before the specified number of bits has been transferred. This prevents the Smart Transceiver watchdog timer from resetting the chip in the event that fewer than the requested number of bits are transferred by the external clock. See Figure 3.30.



Parameter	**Description**	**Typ**
t_{fin}	Function call to data bit out	41.4 µs
t_{ret}	Return from function	19.2 µs
t_{docki}	Data out to input clock and data sampled	4.8 µs
t_{cklo}	Data sampled to clock low sampled	24.0 µs
t_{cklodo}	Clock low sampled to data output	25.8 µs
f	Clock frequency (max)	18.31 kHz

Figure 3.30 Neurowire (SPI) Slave Timing

The algorithm for each bit of output/input for the Neurowire slave objects is described below. In this description, the default active clock edge (positive) is assumed; if the `invert` keyword is used, all clock levels stated should be reversed.

1. Set IO9 to the next output bit value.
2. Test pin IO8, the clock input, for a high level. This is the test for the rising edge of the input clock. If the input clock is still low, sample the timeout event pin and abort if high.
3. When the input clock is high, store the next data input bit as sampled on pin IO10.
4. Test the input clock for a low input level. This is the test for the falling edge of the input clock. If the input clock is still high, sample the timeout event pin and abort if high.
5. When the input clock is low, return to step 1 if there are more bits to be processed.
6. Else return the number of bits processed.

When either clock input test fails (that is, the clock is sampled *before* the next transition), there is an additional timeout check time of 19.8 µs (wait for clock high) or 19.2 µs (wait for clock low) added to that stage of the algorithm.

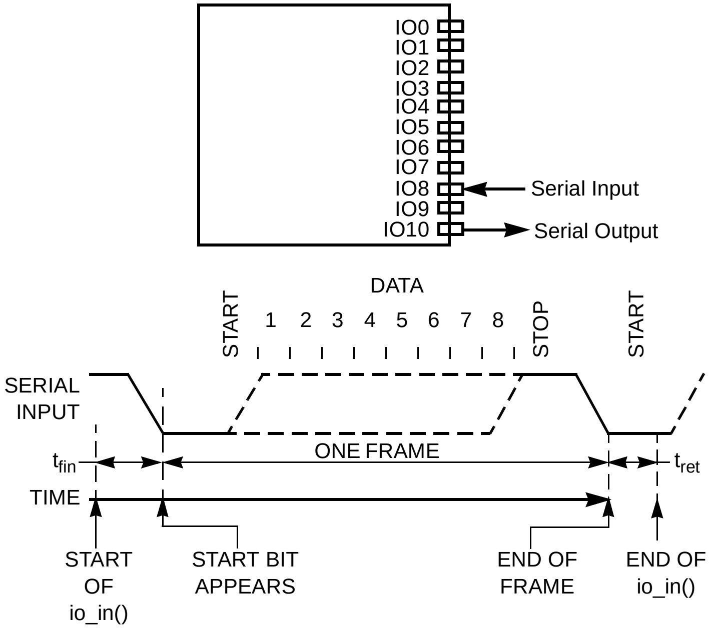
The chip select logic for the Neurowire slave can be handled by the user through a separate bit input object, along with an appropriate handshaking algorithm implemented by the user application program. In order to prevent unnecessary timeouts, the setup and hold times of the chip select line, relative to the start and end of the external clock, must be satisfied.

The timeout input pin can either be connected to an external timer or to an output pin of the Smart Transceiver that is declared as a oneshot object.

3.5.6 Serial Input/Output

Pin IO8 may be configured as an asynchronous serial input line, and pin IO10 may be configured as an asynchronous serial output line. The bit rates for input and for output may be independently specified to be 600, 1200, 2400, or 4800 bits/second at a 10MHz input clock rate. The data rate scales proportionally to the input clock rate. For example, at 40MHz, the bit rates would be 2400, 4800, 9600, or 19,200 bits/second.

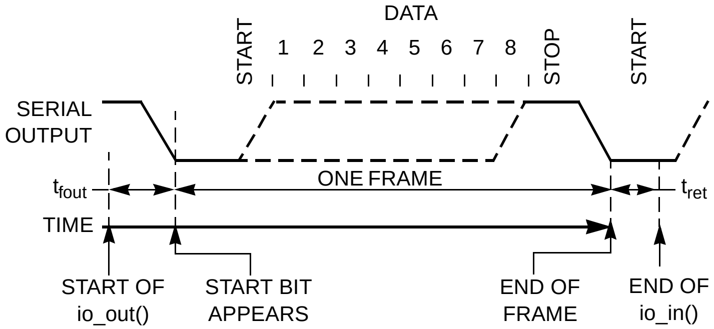
The frame format is fixed at 1 start bit, 8 data bits, and 1 stop bit; and up to 255 bytes may be transferred at a time. Either a serial input or a serial output operation (but not both) may be in effect at any one time. The interface is half-duplex only. This function suspends application processing until the operation is completed. On input, the *io_in()* request will time out after 20 character times if no start bit is received. If the stop bit has the wrong polarity (it should be a 1), the input operation is terminated with an error. The application code can use bit I/O pins for flow control handshaking if required. This function is useful for transferring data to serial devices such as terminals, modems, and computer serial interfaces. See Figures 3.31 and 3.32.



Symbol	Description	Typ @ 10MHz
t_{fin}	Function call to input sample Min (first sample) Max (timeout)	 67 µs 20 byte frame
t_{ret}	Return from function	10 µs

Figure 3.31 Serial Input Object

The duration of this function call is a function of the number of data bits transferred and the transmission bit rate. t_{fin} (max) refers to the maximum amount of time this function will wait for a start bit to appear at the input. After this time, the function will return a 0 as data. t_{fin} (min) is the time to the first sampling of the input pin. As an example, the timeout period at 2400 bits/second is (20 x 10 x 1/ 2400) + t_{fin} (min).



Symbol	Description	Typ @ 10MHz
t_{fout}	Function call to start bit	79 µs
t_{ret}	Return from function	10 µs

Figure 3.32 Serial Output

The duration of this function call is a function of the number of data bits transferred and the transmission bit rate. As an example, to output 100 bytes at 300 bits/second would require a time duration of (100 x 10 x 1/300) + t_{fout} + t_{ret}.

3.5.7 Touch Input/Output

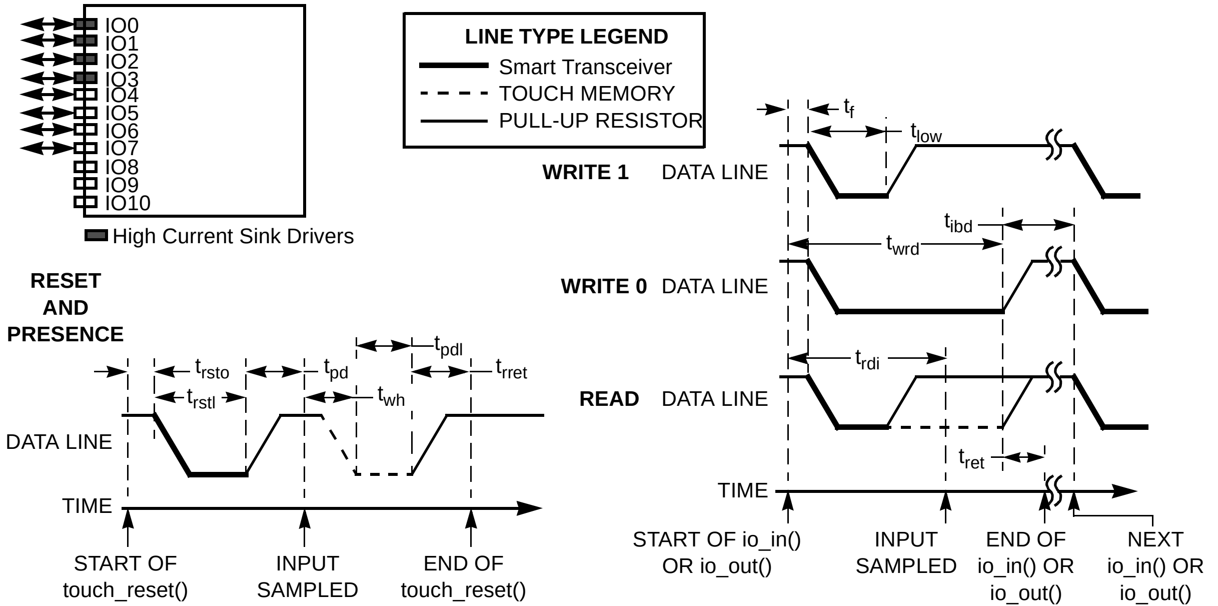
The Touch I/O object enables easy interface to any slave device which adheres to Dallas Semiconductor's Touch Memory™ standard. This interface is a one-wire, open-drain, bidirectional connection.

Up to eight Touch Memory busses may be connected to a Smart Transceiver through the use of the first eight I/O pins, IO0 – IO7. The only additional component required for this is a pull-up resistor on the data line (refer to the Touch Memory specification below on how to select the value of the pull-up resistor). The high current sink capabilities of IO0 – IO3 pins of the Smart Transceiver can be used in applications where long wire lengths are required between the Touch Memory device and the Smart Transceiver.

The slave acquires all necessary power for its operation from the data line. Upon physical connection of a Touch Memory device to a master (in this case the Smart Transceiver), the Touch Memory generates a low presence pulse to inform the master that it is awaiting a command. The Smart Transceiver can also request a presence pulse by sending a reset pulse to the Touch Memory device.

Commands and data are sent bit by bit to make bytes, starting with the LSB. The synchronization between the Smart Transceiver and the Touch Memory devices is accomplished through a negative-going pulse generated by the Smart Transceiver.

Figure 3.33 shows the details of the reset pulse in addition to the read/write bit slots.



Symbol	Description	Min	Typ	Max
t_{rsto}	Reset call to data line low	—	60.0 µs	—
t_{rstl}	Reset pulse width	—	500 µs	—
t_{pdh}	Reset pulse release to data line high 10MHz 5MHz	 4.8 µs 9.6 µs	 — —	 275 µs 275 µs
t_{pdl}	Presence pulse width	—	120.0 µs	—
t_{wh}	Data line high detect to presence pulse	—	80 µs	—
t_{rret}	Return from reset function	—	12.6 µs	—
t_f	I/O call to data line low (start of bit slot)	—	125.4 µs	—
t_{low}	Start pulse width 10MHz 5MHz	 — —	 4.2 µs 8.4 µs	 — —
t_{rdi}	Start pulse edge to sampling of input (read operation) 10MHz 5MHz	 — —	 15.0 µs 18.0 µs	 — —
t_{wrd}	Start pulse edge to Smart Transceiver releasing the data line 10MHz 5MHz	 — —	 66.6 µs 72.0 µs	 — —
t_{ibd}	Inter-bit delay 10MHz 5MHz	 — —	 61.2 µs 122.4 µs	 — —
t_{ret}	Return from I/O call	—	42.6 µs	—

Figure 3.33 Touch I/O Object

The leveldetect input object can be used for detection of asynchronous attachments of Touch Memory devices to the Smart Transceiver. In such a case, the leveldetect input object is overlaid on top of the Touch I/O object. Refer to the *Neuron C Programmer's Guide* for information on I/O object overlays.

The Touch I/O object can run at Smart Transceiver clock rates of 5MHz and 10MHz only. This is because the Touch I/O object is designed to meet the Touch Memory timing specification at those Smart Transceiver clock speeds only. The Touch I/O object is not supported at clock rates faster than 10MHz or slower than 5MHz.

For more specific information on the mechanical, electrical, and protocol specifications, refer to the *Book of DS19xx Touch Memory Standards*, available from Dallas Semiconductor Corporation.

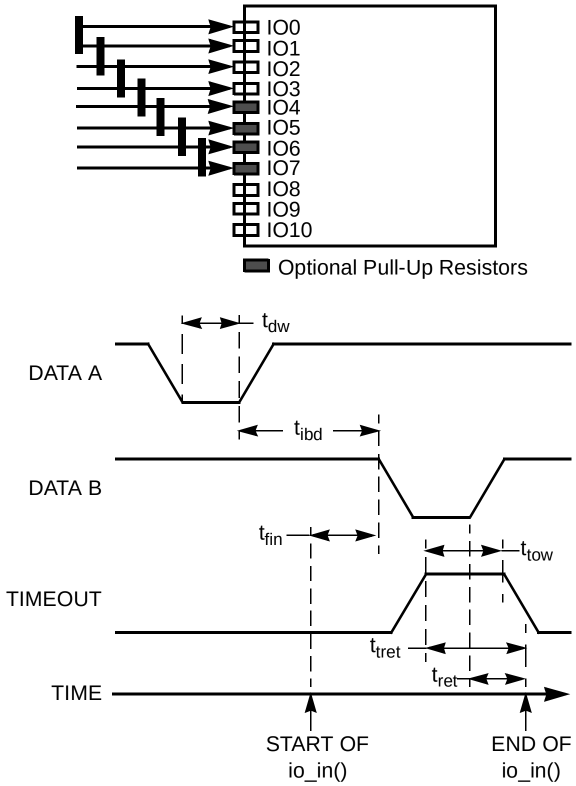
3.5.8 Wiegand Input

This input object provides an easy interface to any card reader supporting the Wiegand standard. Data from the reader is presented to the Smart Transceiver through the use of two of its first eight I/O pins, IO0 – IO7. Up to four Wiegand devices may be connected to the Smart Transceiver. Data is read MSB first.

Wiegand data starts as a negative-going pulse on one of the two pins selected. One input represents a logical 0 bit and the other pin a logical 1, as selected through the I/O declaration. The bit data on the two lines are mutually exclusive and are spaced at least 150 µs apart. Figure 3.34 shows the timing relationship of the two data lines with respect to each other and the Smart Transceiver.

Any unused I/O pin from IO0 to IO7 may be optionally selected as the timeout pin. When the timeout pin goes high, the function aborts and returns. The application processor's watchdog timer is automatically updated during the operation of this input object.

Incoming data on any of the Wiegand input pins is sampled by the Smart Transceiver every 200ns at a 10MHz clock (scales inversely with the clock frequency). Since the Wiegand data is usually asynchronous, care must be taken in the application program to ensure that this function is called in a timely manner in order that no incoming data is lost.



Symbol	**Description**	**Min**	**Typ**	**Max**
t_{fin}	Function call to start of second data edge	—	75.6 µs	—
t_{dw}	Input data width (at 10MHz)	200 ns	100 µs	880 ms
t_{ibd}	Inter-bit delay	150 µs	—	900 µs
t_{tow}	Timeout pulse width	—	39 µs	—
t_{tret}	Timeout to function return	—	18.0 µs	—
t_{ret}	Last data bit to function return	—	74.4 µs	—

Figure 3.34 Wiegand Input Object

3.6 Timer/Counter Input Objects

The FT 3120 and FT 3150 Smart Transceivers have two 16-bit timer/counters. For the first timer/counter, IO0 is used as the output, and a multiplexer selects one of IO4 – IO7 as the input. The second timer/counter uses IO1 as the output and IO4 as the input (see Figure 2.7). Multiple timer/counter input objects may be declared on different pins within a single application. By calling the *io_select()* function, the application can use the first timer/counter in up to four different input functions. If a timer/counter is configured in one of the output functions, or as a quadrature input, then it can not be reassigned to another timer/counter object in the same application program.

The timing numbers shown in this section are valid for both an explicit I/O call or an implicit I/O call through a *when* clause, and are assumed to be for a Smart Transceiver running at 10MHz.

Input timer/counter objects have the advantage (over non-timer/counter objects) in that input events will be captured even if the application processor is occupied doing something else when the event occurs. A true *when* statement condition for an event being measured by a timer/counter is the completion of the measurement and a value being returned to an event register. If the processor is delayed due to software processing and cannot read the register before another event occurs, then the value in the register will reflect the status of the last event. The timer/counters are automatically reset upon completion of a measurement. **The first measured value of a timer/counter is always discarded to eliminate the possibility of a bad measurement after the chip comes out of a reset condition.** Single events can not be measured with the timer/counters.
Figure 3.35 shows an example of how the timer/counter objects are processed with a Neuron C *when* statement.

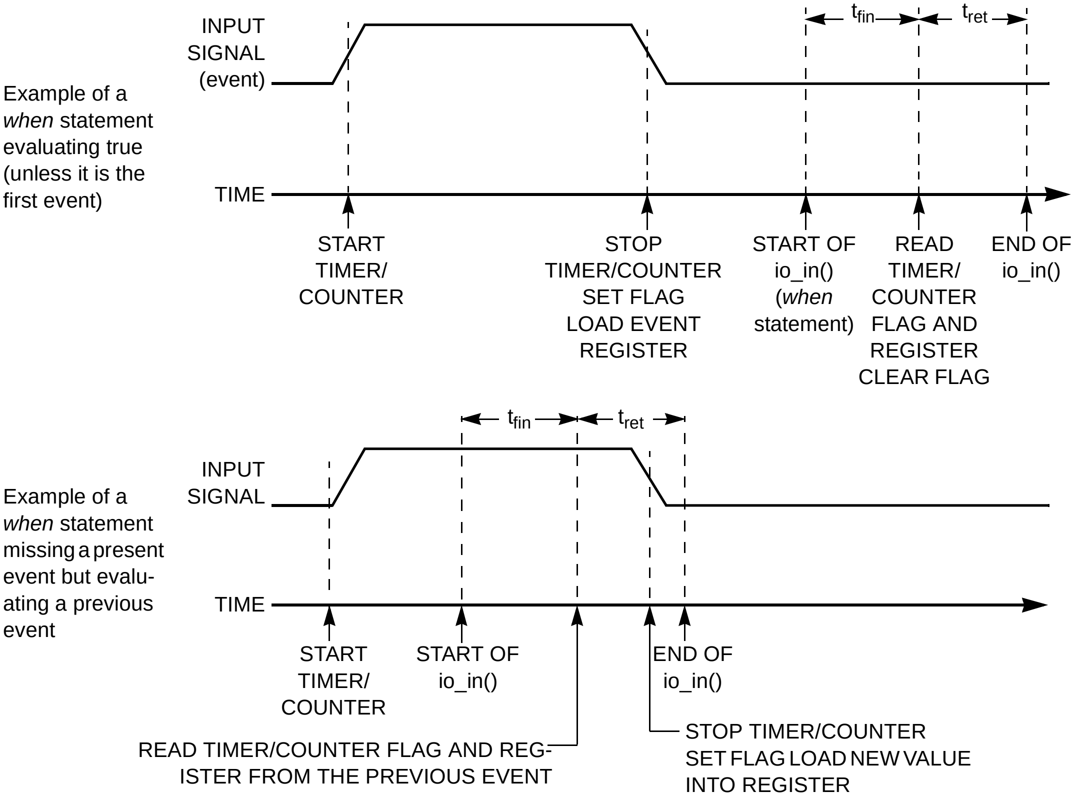
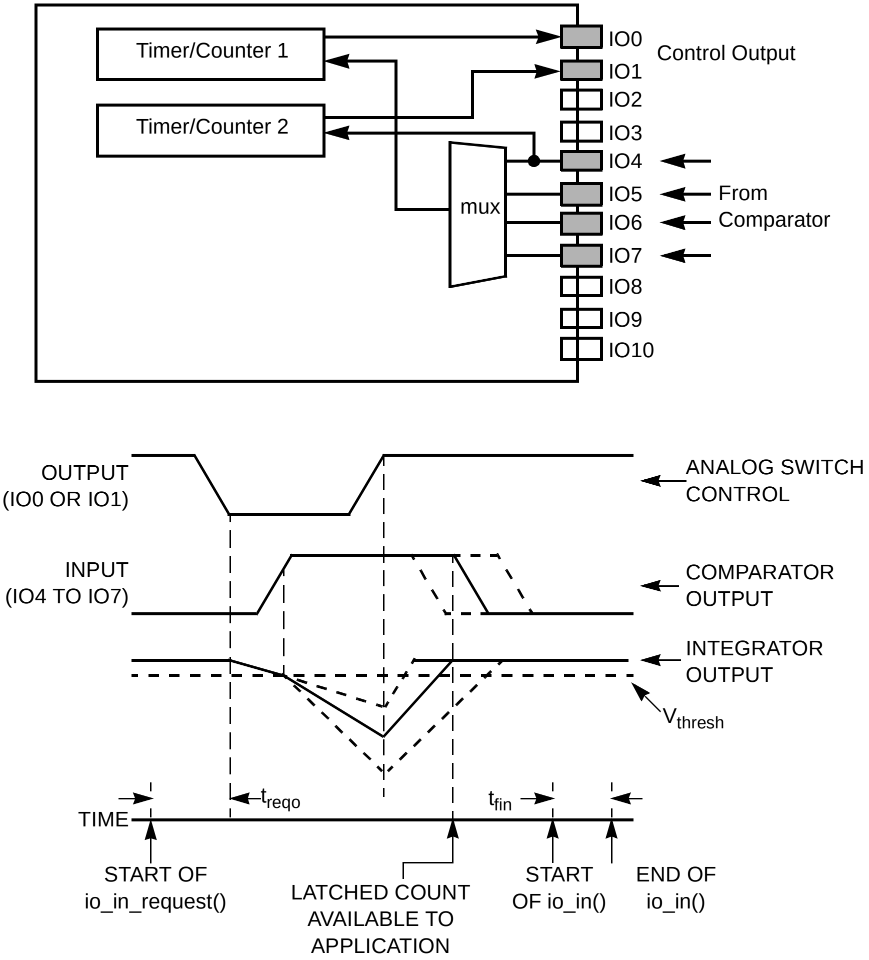

Example of a *when* statement evaluating true (unless it is the first event)
INPUT SIGNAL (event)
TIME
t_{fin}
t_{ret}
START TIMER/ COUNTER
STOP TIMER/COUNTER SET FLAG LOAD EVENT REGISTER
START OF io_in() (*when* statement)
READ TIMER/ COUNTER FLAG AND REGISTER CLEAR FLAG
END OF io_in()
Example of a *when* statement missing a present event but evaluating a previous event
INPUT SIGNAL
TIME
t_{fin}
t_{ret}
START TIMER/ COUNTER
START OF io_in()
END OF io_in()
READ TIMER/COUNTER FLAG AND REGISTER FROM THE PREVIOUS EVENT
STOP TIMER/COUNTER SET FLAG LOAD NEW VALUE INTO REGISTER

Figure 3.35 Example of *when* Statement Processing Using the Ontime Input Function

3.6.1 Dualslope Input

This input object uses a timer/counter to control and measure the integration periods of a dualslope integrating analog to digital converter (see Figure 3.36). The timer/counter provides the control output signal and senses a comparator output signal. The control output signal controls an external analog multiplexer which switches between the unknown input voltage and a voltage reference. The timer/counter's input pin is driven by an external comparator which compares the integrator's output with a voltage reference. At the end of conversion, the external comparator will drive a low level to one of pins IO4 – IO7. If external circuitry indicates "end of conversion" with a high level, use the `invert` keyword in the I/O declaration.

The resolution and range of the timer/counter period options is shown by Table 3.6 in section 3.8, *Notes*, at the end of this chapter.



Symbol	**Description**	**Min**	**Typ**	**Max**
t_{reqo}	*io_in_request()* to output toggle	—	75.6 µs	—
t_{fin}	Input function call and return	—	82.8 µs	—

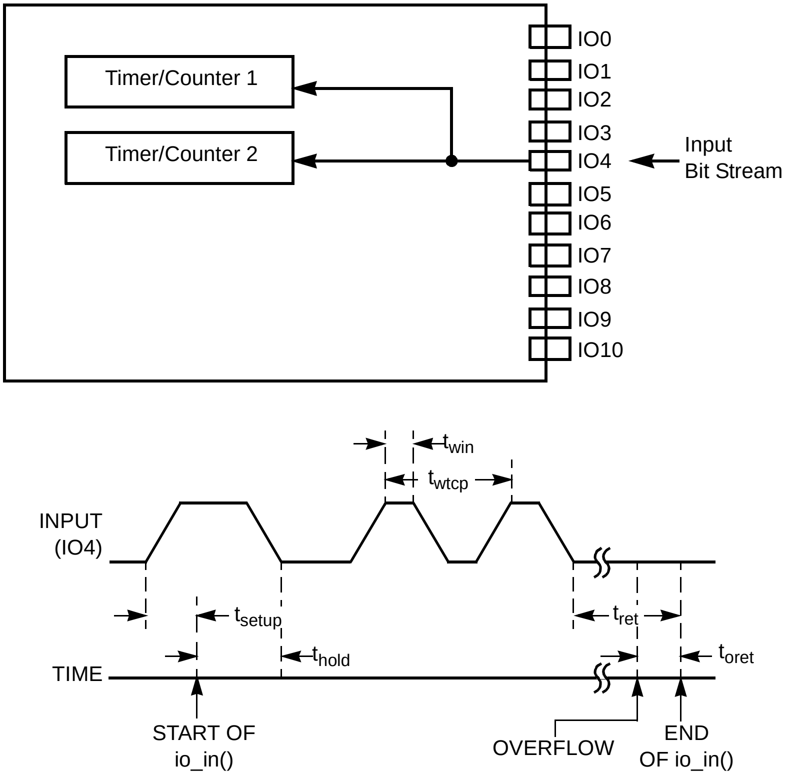
Figure 3.36 Dualslope Input Object

3.6.2 Edgelog Input

The edgelog input object can record a stream of input pulses measuring the consecutive low and high periods at the input and storing them in user-defined storage (see Figure 3.37). The values stored represent the units of clock period between rising and falling input signal edges. Both timer/counters of the Smart Transceiver are used for this object.

The measurement series starts on the first rising (positive) edge, unless the *invert* keyword is used in the I/O object declaration. The measurement process stops whenever an overflow condition is sensed on either timer/counter.

The resolution and range of the timer/counter period options are shown in Table 3.6 in section 3.8, *Notes*. This object is useful for analyzing an arbitrarily-spaced stream of input edges (or pulses), such as the output of a UPC bar-code reader or infrared receiver.



Symbol	**Description**	**Min**	**Typ**	**Max**
t_{setup}	Input data setup	0	—	—
t_{win}	Input pulse width	1 T/C clk	—	65,534 T/C clks
t_{hold}	*io_in()* call to data input edge for inclusion of that pulse	26.4 µs	—	—
t_{wtcp}	Two consecutive pulse widths	104 µs	—	—
t_{oret}	Return on overflow	—	42.6 µs	—
t_{ret}	Return on count termination	—	49.6 µs	—

Note: T/C clk represents the period of the clock used during the declaration of the I/O object.

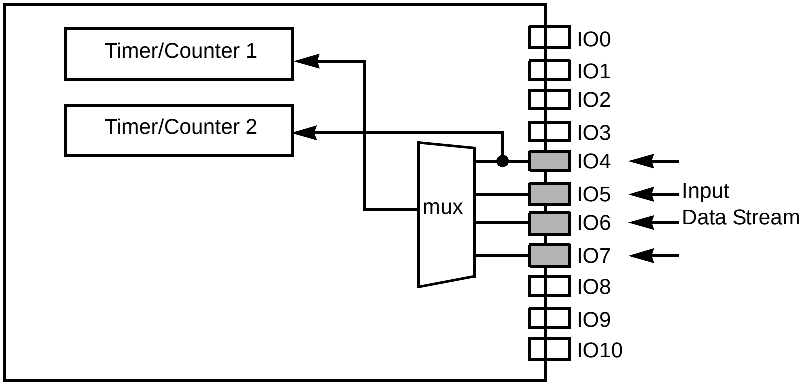
Figure 3.37 Edgelog Input Object

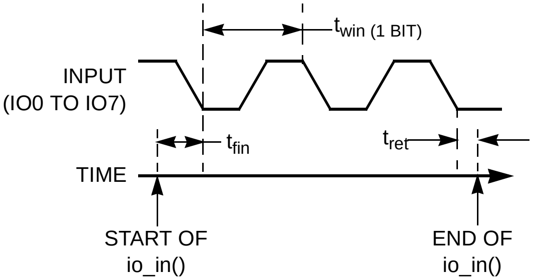
3.6.3 Infrared Input

The infrared input object is used to capture a stream of data generated by a class of infrared remote control devices (see Figure 3.38). The input to the object is the demodulated series of bits from infrared receiver circuitry. The period of the on/off cycle determines the data bit value, a shorter cycle indicating a one, and a longer cycle indicating a zero. The actual threshold for the on/off determination is set at the time of the call of the function. The measurements are made between the negative edges of the input bits unless the *invert* keyword is used in the I/O declaration.

The infrared input object, based on the input data stream, generates a buffer containing the values of the bits received. The resolution and range of the timer/counter period options is shown in Table 3.6 in section 3.8, *Notes*, at the end of this chapter.

This function can be used with an off-the-shelf IR demodulator such as an NEC µPD1913 or Sharp GP1U50X to quickly develop an infrared interface to the Smart Transceiver. The edgelog input object can also be used for this purpose. However, this requires more code.





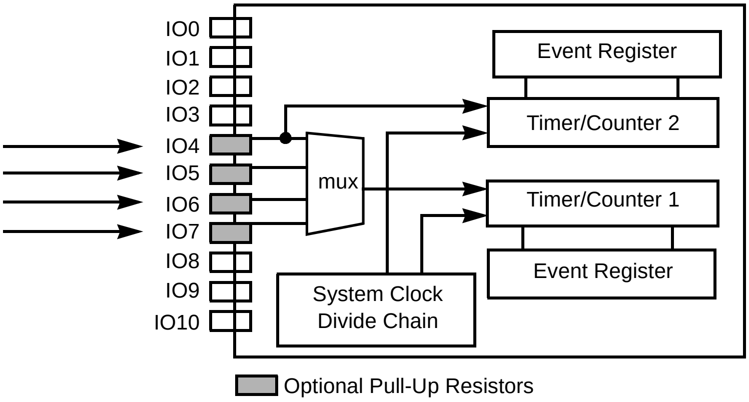
Symbol	Description	Min	Typ	Max
t_{fin}	Function call to start of input sampling	—	82.2 µs	—
t_{ret}	End of last valid bit to function return	max-period	max-period	—
t_{win}	Minimum input period width	—	93 µs	—

Note: max-period is the timeout period passed to the function at the time of the call.

Figure 3.38 Infrared Input Object

3.6.4 Ontime Input

A timer/counter may be configured to measure the time for which its input is asserted. Table 3.6 shows the resolution and maximum times for different I/O clock selections. Assertion may be defined as either logic high or logic low. This object may be used as a simple analog-to-digital converter with a voltage-to-time circuit, or for measuring velocity by timing motion past a position sensor. See Figures 3.35 and 3.39.



Reference Figure 3.35

Symbol	Description	Typ @ 10MHz
t_{fin}	Function call to input sample	86 µs
t_{ret}	Return from function	52/22 µs*

*If the measurement is new, t_{ret} = 52 µs. If a new time is not being returned, t_{ret} = 22 µs.

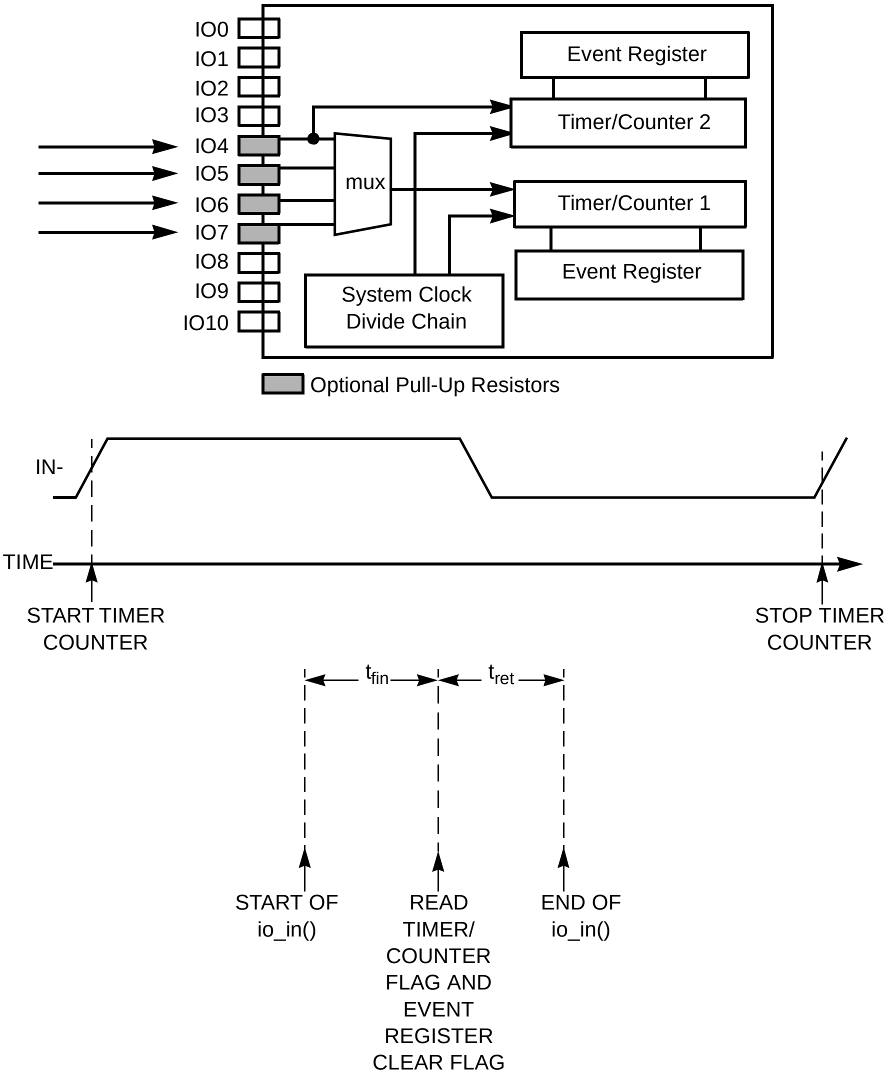
Figure 3.39 Ontime Latency Values

This is a level-sensitive function. The active level of the input signal gates the clock driving the internal counter in the Smart Transceiver.

The actual active level of the input depends on whether or not the invert option was used in the declaration of the function block. The default is the high level.

3.6.5 Period Input

A timer/counter may be configured to measure the period from one rising or falling edge to the next corresponding edge on the input. Table 3.6 shows the resolution and maximum time measured for various clock selections. This object is useful for instantaneous frequency or tachometer applications. Analog-to-digital conversion can be implemented using a voltage-to-frequency converter with this object. See Figure 3.40.



Reference Figure 3.35

Symbol	**Description**	**Typ @ 10MHz**
t_{fin}	Function call to input sample	86 µs
t_{ret}	Return from function	52/22 µs*

Figure 3.40 Period Input Latency Values

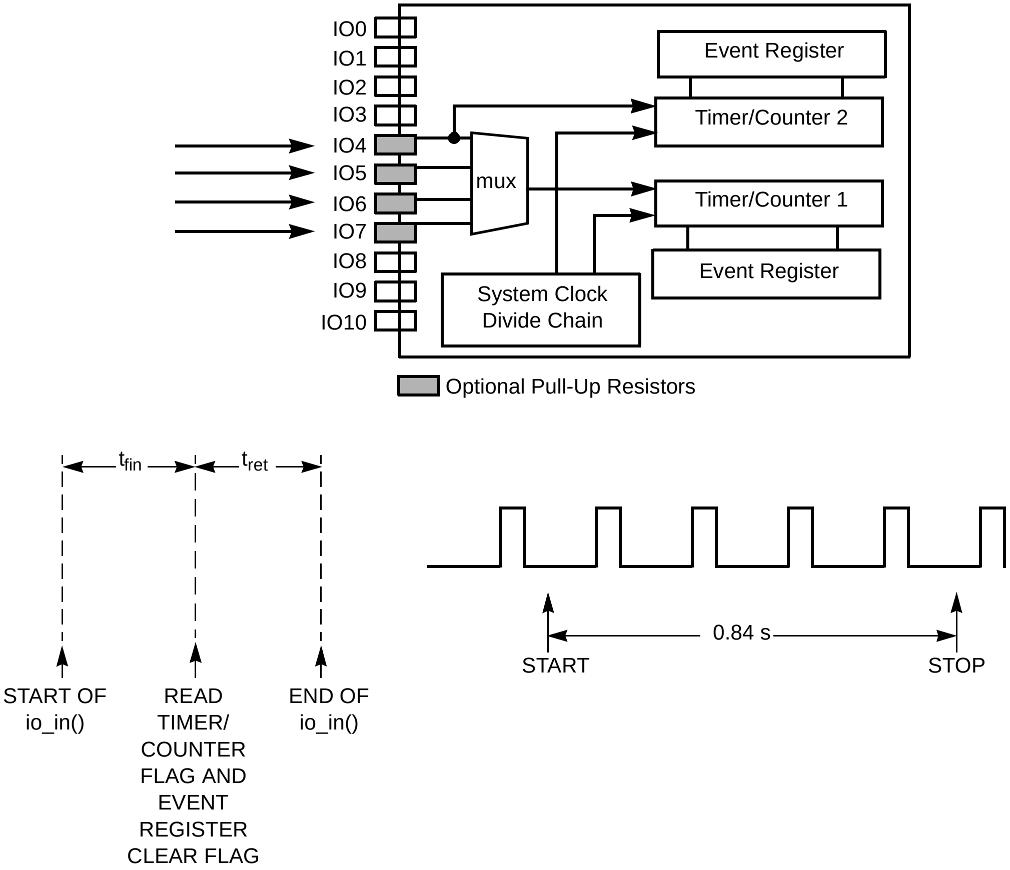
This is an edge-sensitive function. The clock driving the internal counter in the Smart Transceiver is free running. The detection of active input edges stops and resets the counter each time.

The actual active edge of the input depends on whether or not the invert option was used in the declaration of the function block. The default is the negative edge.

Since the period function measures the delay between two consecutive active edges, the invert option has no effect on the returned value of the function for a repeating input waveform.

3.6.6 Pulsecount Input

A timer/counter may be configured to count the number of input edges (up to 65,535) in a fixed time (0.8388608 second) at all allowed input clock rates. Edges may be defined as rising or falling. This object is useful for average frequency measurements, or tachometer applications. See Figure 3.41.



Reference Figure 3.35

Symbol	Description	Typ @ 10MHz
t_{fin}	Function call to input sample	86 µs
t_{ret}	Return from function	52/22 µs*

*If the measurement is new, t_{ret} = 52 µs. If a new time is not being returned, t_{ret} = 22 µs.

Figure 3.41 Pulse Count Input Latency Values

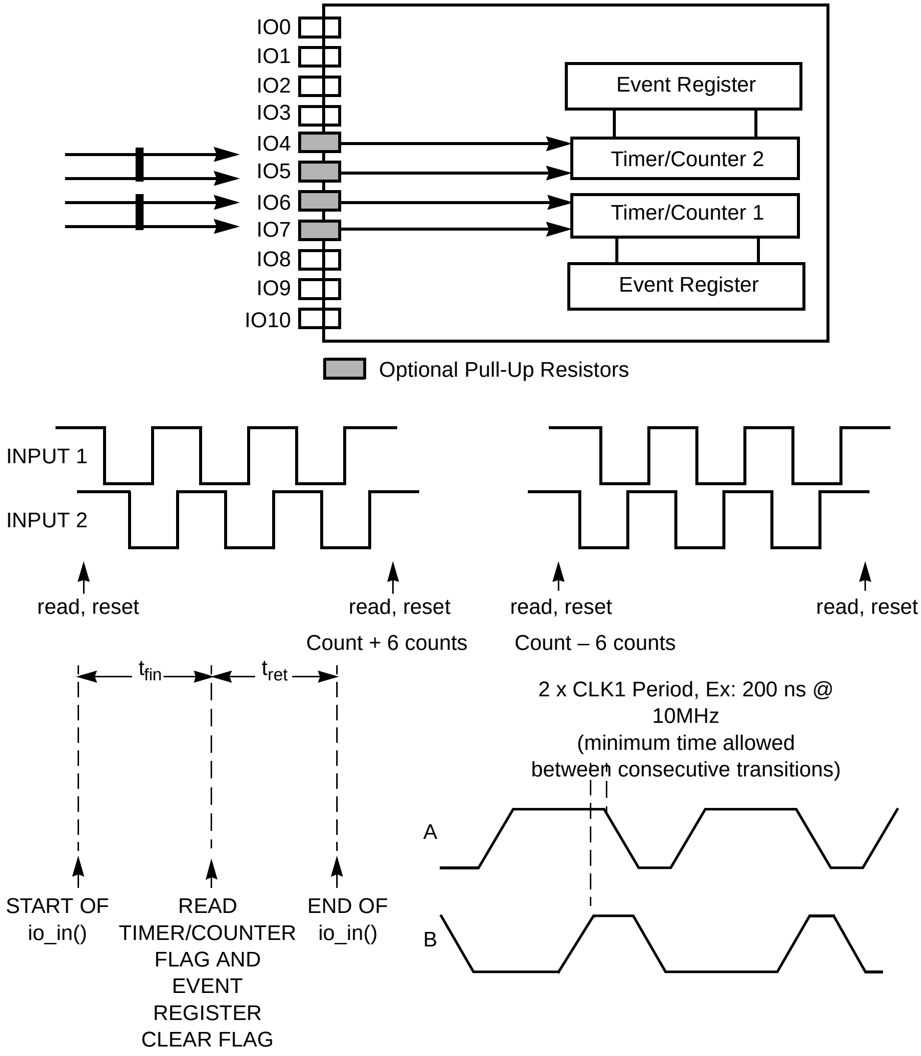
This is an edge-sensitive function. The clock driving the internal counter in the Smart Transceiver is the actual input signal. The counter is reset automatically every 0.839 second.

The internal counter increments with every occurrence of an active input edge. Every 0.839 second, the content of the counter is saved and the counter is then reset to 0. This sequence is repeated indefinitely.

The actual active edge of the input depends on whether or not the invert option was used in the declaration of the function block. The default is the negative edge.

3.6.7 Quadrature Input

A timer/counter may be configured to count transitions of a binary Gray code input on two adjacent input pins. The Gray code is generated by devices such as shaft encoders and optical position sensors which generate the bit pattern (00,01,11,10,00, ...) for one direction of motion and the bit pattern (00,10,11,01,00, ...) for the opposite direction. Reading the value of a quadrature object gives the arithmetic net sum of the number of transitions since the last time it was read (– 16,384 to 16,383). The maximum frequency of the input is one-quarter of the input clock rate, for example 2.5MHz at 10MHz Smart Transceiver input clock. Quadrature devices may be connected to timer/counter 1 via pins IO6 and IO7, and timer/counter 2 via pins IO4 and IO5. See Figure 3.42. If the second input transitions low while the first input is low and high while the first input is high, the counter counts up. Otherwise, the count is down.



Reference Figure 3.35

Symbol	Description	Typ @ 10MHz
t_{fin}	Function call to input sample	90 µs
t_{ret}	Return from function	88 µs

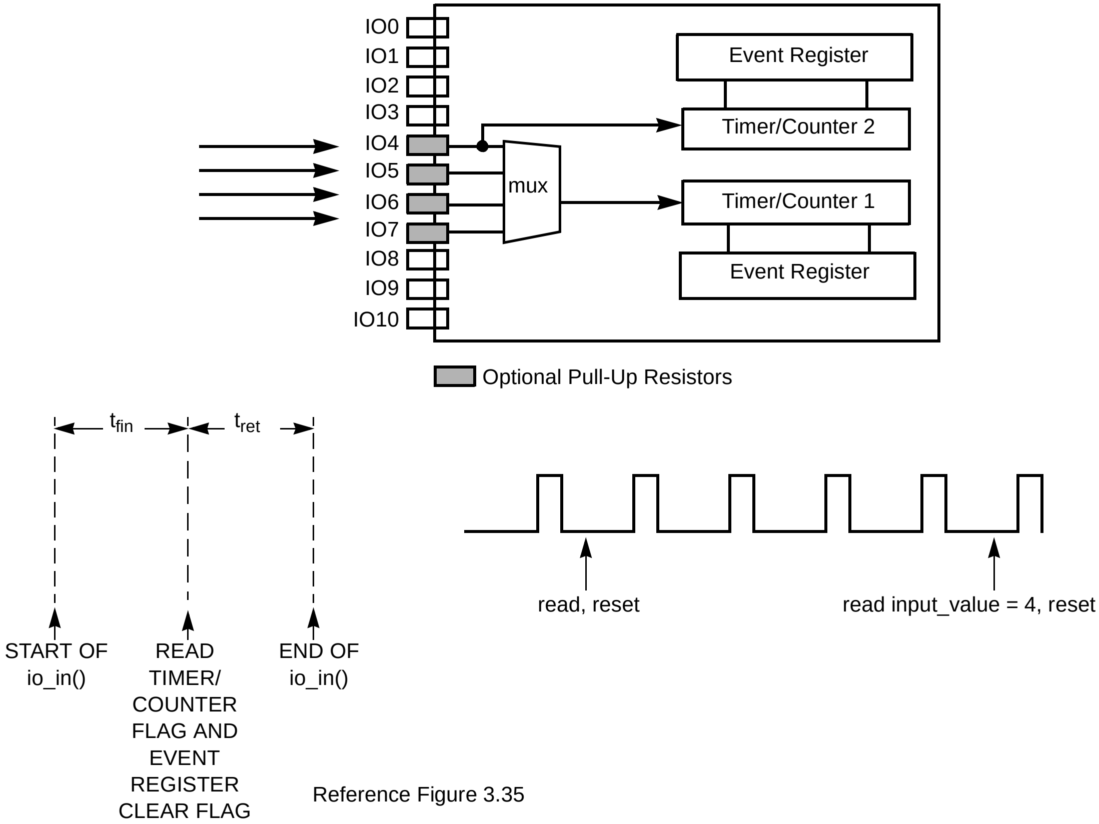
Figure 3.42 Quadrature Input Latency Values

A call to this function returns the current value of the quadrature count since the last read operation. The counter is then reset and ready for the next series of input transitions. The count returned is a 16-bit signed binary number, capped at ±16K.

The number shown in the diagram above is the minimum time allowed between consecutive transitions at either input of the quadrature function block. For more information, see the, *Neuron Chip Quadrature Input Function Interface* engineering bulletin.

3.6.8 Totalcount Input

A timer/counter may be configured to count either rising or falling input edges, but not both. Reading the value of a totalcount object gives the number of transitions since the last time it was read (0 to 65,535). Maximum frequency of the input is one-quarter of the input clock rate, for example 2.5MHz at a maximum of 10MHz Smart Transceiver input clock. This object is useful for counting external events such as contact closures, where it is important to keep an accurate running total. See Figure 3.43.



Symbol	Description	Typ @ 10MHz
t_{fin}	Function call to input sample	92 µs
t_{ret}	Return from function	61 µs

Figure 3.43 Totalcount Input Latency Values

A call to this function returns the current value of the totalcount value corresponding to the total number of active clock edges since the last call. The counter is then reset, and ready for the next series of input transitions.

The actual active edge of the input depends on whether or not the invert option was used in the declaration of the function block. The default is the negative edge.

3.7 Timer/Counter Output Objects

3.7.1 Edgedivide Output

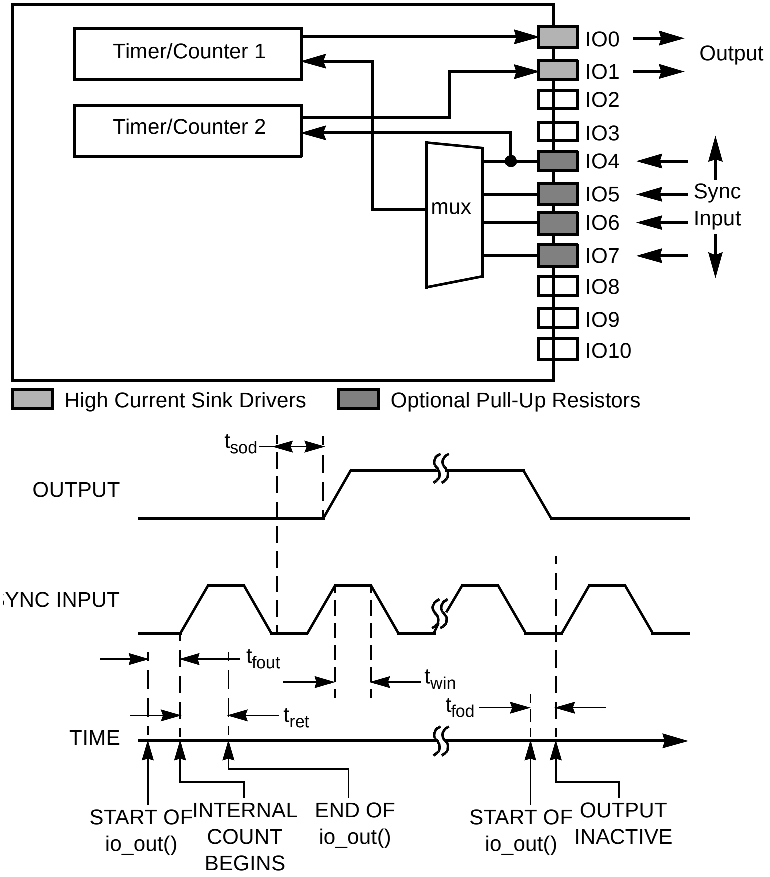
This output object acts as a frequency divider by providing an output frequency on either pin IO0 or IO1. The output frequency is a divided-down version of the input frequency applied on pins IO4 – IO7. The object is useful for any divide-by-n operation, where *n* is passed to the timer/counter object through the application program and can be from 1 to 65,535. The value of 0 forces the output to the off level and halts the timer/counter.

A new divide value will not take effect until after the output toggles, with two exceptions: if the output is initially disabled, the new (non-zero) output will start immediately after t_{fout}; or, for a new divide value of 0, the output is disabled immediately.

Normally the negative edges of the input sync pulses are the active edge. Using the *invert* keyword in the object declaration makes the positive edge active.

The initial state of the output pin is logic 0 by default. This can also be changed to logic 1 through the object declaration.

Figure 3.44 shows the pinout and timing information for this output object.

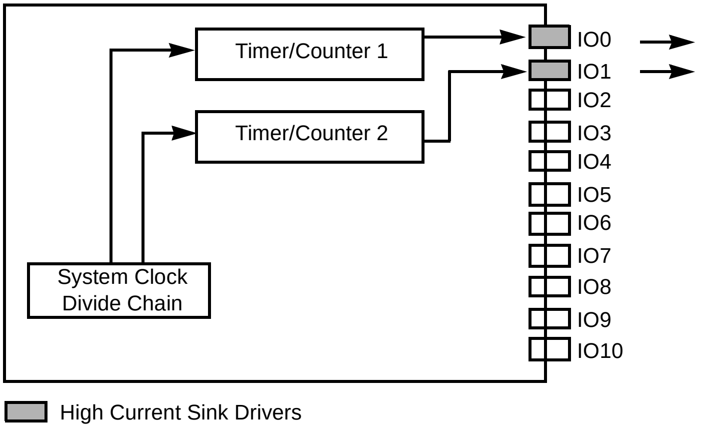


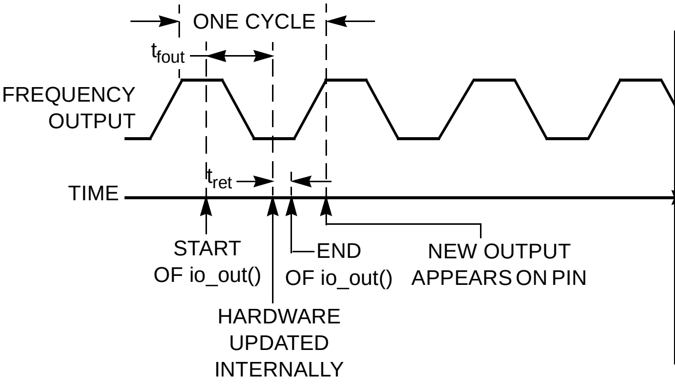
Symbol	**Description**	**Min**	**Typ**	**Max**
t_{fout}	Function call to start of timer	—	96 µs	—
t_{fod}	Function to output disable	—	82.2 µs	—
t_{sod}	Active sync edge to output toggle	550 ns	—	750 ns
t_{win}	Sync input pulse width (10MHz)	200 ns	—	—
t_{ret}	Return from function	—	13 µs	—

Figure 3.44 Edgedivide Output Object

3.7.2 Frequency Output

A timer/counter may be configured to generate a continuous square wave of 50% duty cycle. Writing a new frequency value to the device takes effect at the end of the current cycle. This object is useful for frequency synthesis to drive an audio transducer, or to drive a frequency to voltage converter to generate an analog output. See Figure 3.45.





Frequency Resolution and Maximum Range at 10MHz

CLK	Resolution	Range	Unit
0	0.4	26.21	µs
1	0.8	52.42	µs
2	1.6	104.86	µs
3	3.2	209.71	µs
4	6.4	419.42	µs
5	12.8	838.85	µs
6	25.6	1677	µs
7	51.2	3355	µs

Symbol	**Description**	**Typ @ 10MHz**
t_{fout}	Function call to output update	96 µs
t_{ret}	Return from function	13 µs

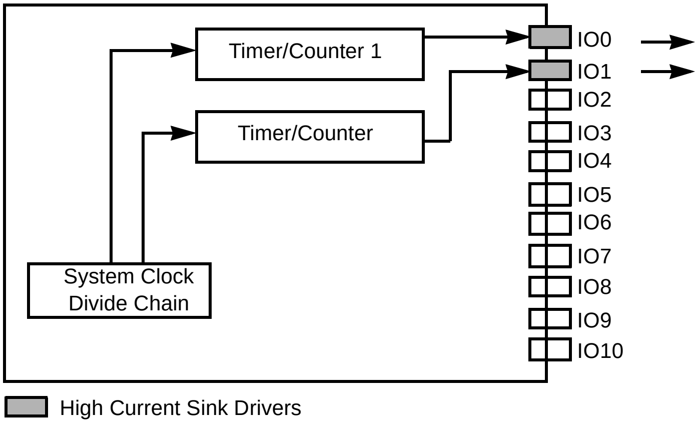
Figure 3.45 Frequency Output Latency Values

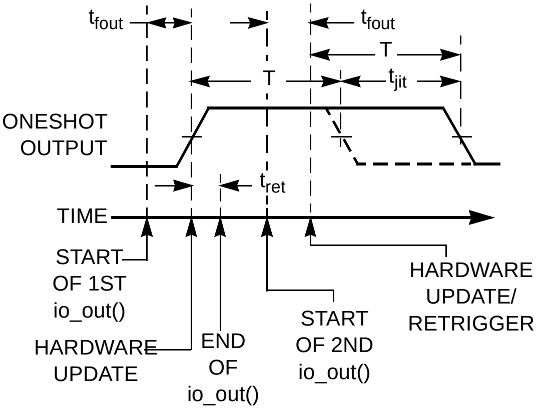
A new frequency output value will not take effect until the end of the current cycle. There are two exceptions to this rule. If the output is disabled, the new (non-zero) output will start immediately after t_{fout}. Also, for a new output value of zero, the output is disabled immediately and not at the end of the current cycle.

A disabled output is a logic zero by default unless the *invert* keyword is used in the I/O object declaration. The resolution and range for this object scale with Smart Transceiver input clock rate, for example: resolution from 0.1 to 12.8 µs and range from 6.55 to 839 ms at 40MHz.

3.7.3 Oneshot Output

A timer/counter may be configured to generate a single pulse of programmable duration. The asserted state may be either logic high or logic low. Retriggering the oneshot before the end of the pulse causes it to continue for the new duration. Table 3.6 in section 3.8, *Notes*, gives the resolution and maximum time of the pulse for various clock selections. This object is useful for generating a time delay without intervention of the application processor. See Figure 3.46.





T = User-defined oneshot output period

Symbol	Description	Typ @ 10MHz	Max
t_{fout}	Function call to output update	96 µs	—
t_{ret}	Return from function	13 µs	—
t_{jit}	Output duration jitter	—	1 timer/counter clock period*

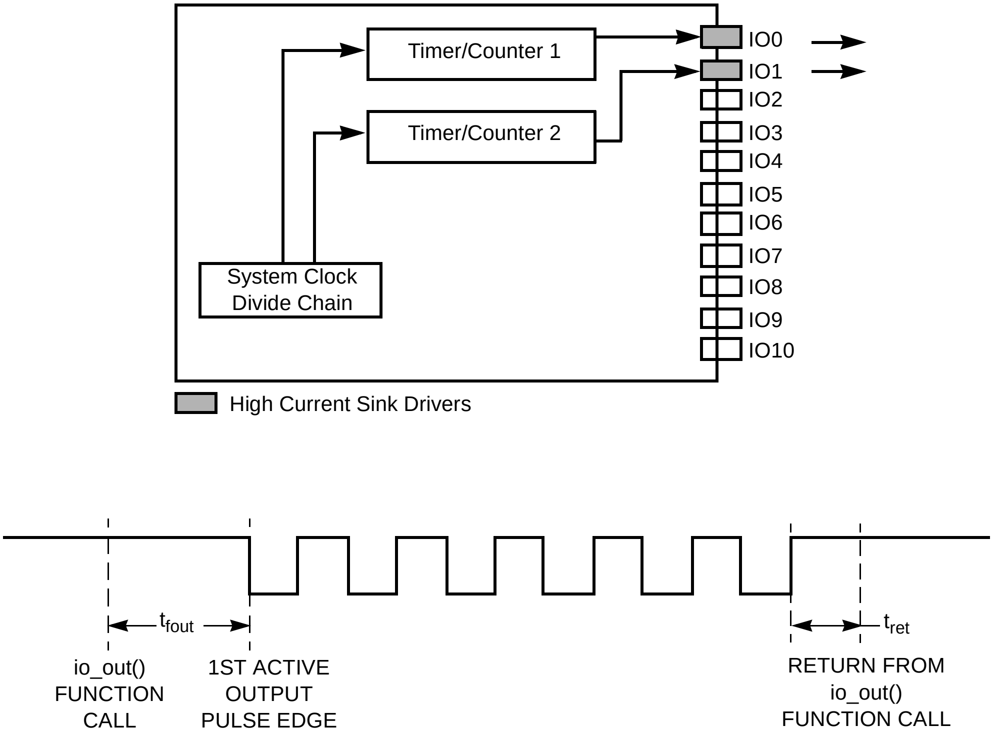
*Timer/counter clock period = (2000ns * 2^(clock))/(input clock in MHz).

Figure 3.46 Oneshot Output Latency Values

While the output is still active, a subsequent call to this function will cause the update to take effect immediately, extending the current cycle. This is, therefore, a retriggerable oneshot function.

3.7.4 Pulsecount Output

A timer/counter may be configured to generate a series of pulses. The number of pulses output is in the range 0 to 65,535, and the output waveform is a square wave of 50% duty cycle. This function suspends application processing until the pulse train is complete. The frequency of the waveform may be one of eight values given by Table 3.7 in section 3.8, *Notes* with clock select values of 0 through 7. This object is useful for external counting devices that can accumulate pulse trains, such as stepper motors. See Figure 3.47.


Timer/Counter 1
Timer/Counter 2
System Clock Divide Chain
IO0
IO1
IO2
IO3
IO4
IO5
IO6
IO7
IO8
IO9
IO10
High Current Sink Drivers
t_{fout}
t_{ret}
io_out() FUNCTION CALL
1ST ACTIVE OUTPUT PULSE EDGE
RETURN FROM io_out() FUNCTION CALL

Symbol	Description	Typ @ 10MHz
t_{fout}	Function call to first active output pulse edge	115 µs
t_{ret}	Return from function	5 µs

Figure 3.47 Pulsecount Output

The return from this function does not occur until all output pulses have been produced.

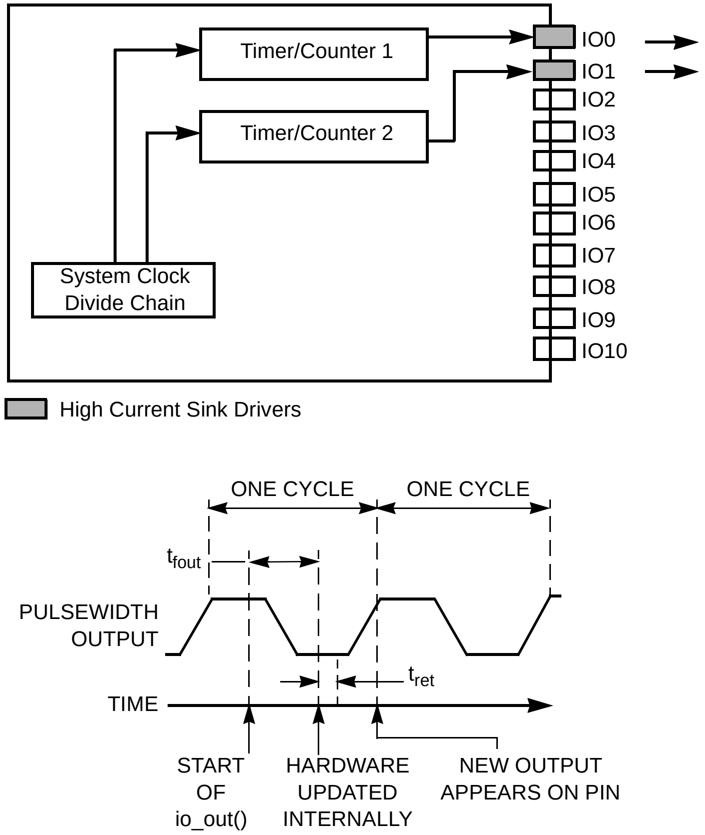
t_{fout} is the time from function call to first output pulse. Therefore, the calling of this function ties up the application processor for a period of N x (pulse period) + t_{fout} + t_{ret}, where N is the number of specified output pulses.

The polarity of the output depends on whether or not the invert option was used in the declaration of the function block. The default is low with high pulses.

3.7.5 Pulsewidth Output

A timer/counter may be configured to generate a pulsewidth modulated repeating waveform. In pulsewidth short function, the duty cycle ranges from 0% to 100% (0/256 to 255/256) of a cycle in steps of about 0.4% (1/256). The frequency of the waveform may be one of eight values given by Table 3.7.

In pulsewidth long function, the duty cycle ranges from 0% to almost 100% (0/65,536 to 65,535/65,536) of a cycle in steps of 15.25 ppm (1/65,536). The frequency of the waveform may be one of eight values given by Table 3.8 in section 3.8, *Notes*. The asserted state of the waveform may be either logic high or logic low. Writing a new pulsewidth value to the device takes effect at the end of the current cycle. A pulsewidth modulated signal provides a simple means of digital-to-analog conversion. See Figure 3.48.



Symbol	Description	Typ @ 10MHz
t_{fout}	Function call to output update	101 µs
t_{ret}	Return from function	13 µs

Figure 3.48 Pulsewidth Output Latency Values

The new output value will not take effect until the end of the current cycle. There are two exceptions to this rule. If the output is disabled, the new (non-zero) output will start immediately after t_{fout}. Also, for a new output value of zero, the output is disabled immediately and not at the end of the current cycle.

A disabled output is a logic 0 by default unless the *invert* keyword is used in the I/O object declaration.

3.7.6 Triac Output

On the FT 3150 and FT 3120 Smart Transceiver, a timer/counter may be configured to control the delay of an output signal with respect to a synchronization input. This synchronization can occur on the rising edge, the falling edge, or both the rising and falling edges of the input signal. For control of AC circuits using a triac device, the sync input is typically a zero-crossing signal, and the pulse output is the triac trigger signal. Table 3.6 shows the resolution and maximum range of the delay. See Figure 3.49.

The output gate pulse is gated by an internal clock with a constant period of 25.6 µs (independent of the Smart Transceiver input clock). Since the input trigger signal (zero crossing) is asynchronous relative to this internal clock, there is a jitter, t_{jit}, associated with the output gate pulse.

The actual active edge of the sync input and the triac gate output can be set by using the clock edge or invert parameters, respectively.

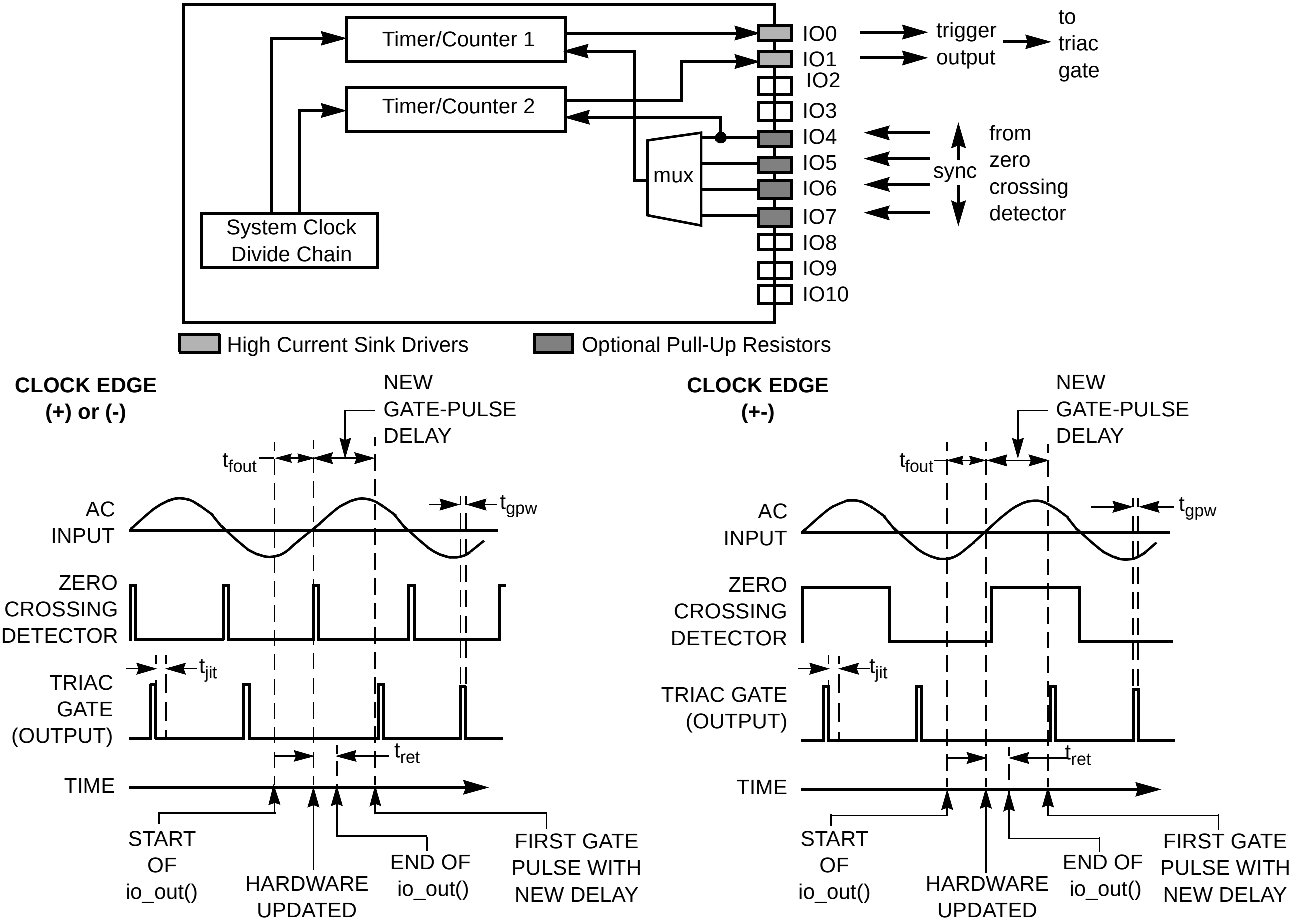


Figure 3.49 Triac Output Latency Values

The hardware update does not happen until the occurrence of an external active sync clock edge. The internal timer is then enabled and a triac gate pulse is generated after the user-defined period has elapsed. This sequence is repeated indefinitely until another update is made to the triac gate pulse delay value.

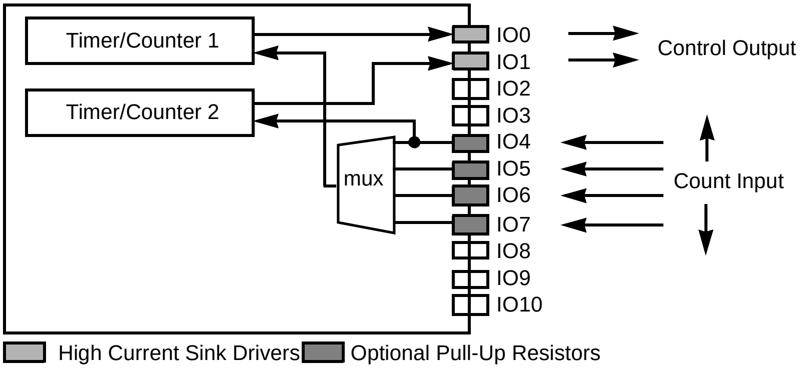
t_{fout} (min) refers to the delay from the initiation of the function call to the first sampling of the sync input. In the absence of an active sync clock edge, the input is repeatedly sampled for 10ms (1/2 wave of a 50 Hz line cycle time), t_{fout} (max), during which the application processor is suspended.

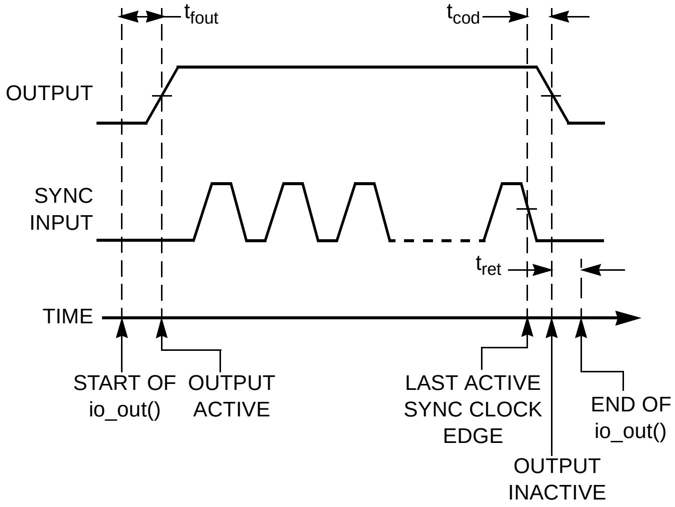
The output gate pulse is gated by an internal clock with a constant period of 25.6µs (independent of the Smart Transceiver input clock). Since the input trigger signal (zero crossing) is asynchronous relative to this internal clock, there is a jitter, t_{jit}, associated with the output gate pulse.

The actual active edge of the sync input and the triac gate output can be set by using the clock edge or invert parameters, respectively.

3.7.7 Triggered Count Output

A timer/counter may be configured to generate an output pulse that is asserted under program control, and de-asserted when a programmable number of input edges (up to 65,535) has been counted on an input pin (IO4 – IO7). Assertion may be either logic high or logic low. This object is useful for controlling stepper motors or positioning actuators which provide position feedback in the form of a pulse train. The drive to the external device is enabled until it has moved the required distance, and then the device is disabled. See Figure 3.50.





Symbol	**Description**	**Typ @ 10MHz**
t_{fout}	Function call to output pulse	109 µs
t_{cod}	Last negative sync Clock edge to output inactive	min 550 ns max 750 ns
t_{ret}	Return from function	7 µs

Figure 3.50 Triggered Count Output Latency Values

The active output level depends on whether or not the invert option was used in the declaration of the function block. The default is high.

3.8 Notes

Various combinations of I/O pins may be configured as basic inputs or outputs. The application program may optionally specify the initial values of basic outputs. Pins configured as outputs may also be read as inputs, returning the value last written.

The gradient behavior of the timing numbers for different Smart Transceiver pins for some of the I/O objects is due to the shift-and-mask operation performed by the Neuron firmware.

For dualslope input, edgelog input, ontime input, and period input, the timer/counter returns a value (or a table of values, in the case of edgelog input) in the range 0 to 65,535, representing elapsed times from 0 up to the maximum range given in Table 3.6.

For ontime input, period input, dualslope, edgelog, and infrared; the timer/counter returns a number in the range 0 to 65,535, representing elapsed times from 0 up to the maximum range given in Table 3.6.

For oneshot output, frequency output, and triac output; the timer/counter may be programmed with a number in the range 0 to 65,535. This number represents the waveform ontime for oneshot output, the waveform period for frequency output, and the control period from sync input to pulse/level output for the triac output. Table 3.6 gives the range and resolution for these timer/counter objects at 10MHz. The clock select value is specified in the declaration of the I/O object in the Neuron C application program, and may be modified at runtime.

Table 3.6 Timer/Counter Resolution and Maximum Range

Clock Select	Oneshot and Triac Outputs; Dualslope, Edgelog, Ontime, and Period Inputs		Frequency Output	
	Resolution (µs)	Maximum Range (ms)	Resolution (µs)	Maximum Range (ms)
0	0.2	13.1	0.4	26.2
1	0.4	26.2	0.8	52.4
2	0.8	52.4	1.6	105
3	1.6	105	3.2	210
4	3.2	210	6.4	419
5	6.4	419	12.8	838
6	12.8	839	25.6	1,678
7	25.6	1,678	51.2	3,355

This table is for a 10MHz input clock. Scale appropriately for other clock rates:
Resolution (µs) = $2^{(\text{Clock Select} + n)}$/(Input Clock in MHz)
Maximum Range (µs) = 65535 x Resolution (µs) x n
n = 1 for oneshot and triac output, and dualslope, edgelog, ontime, and period input
n = 2 for frequency output.
For 20MHz operation, the numbers in this table would be half the value shown.

For pulsewidth short output and pulsecount output, Table 3.7 gives the possible choices for pulsetrain repetition frequencies. Pulsecount can not be used with clock select 0.

Table 3.7 Timer/Counter Square Wave Output

Clock Select (System Clock ÷)	Repetition Rate (Hz)	Repetition Period (µs)	Resolution of Pulse (µs)
0 (÷1) (5MHz)	19,531	51.2	0.2
1 (÷ 2) (2.5MHz)	9,766	102.4	0.4
2 (÷ 4) (1.25MHz)	4,883	204.8	0.8
3 (÷ 8) (625 kHz)	2,441	409.6	1.6
4 (÷ 16) (312.5 kHz)	1,221	819.2	3.2
5 (÷ 32) (156.25 kHz)	610	1,638.4	6.4
6 (÷ 64) (78.125 kHz)	305	3,276.8	12.8
7 (÷ 128) (39.06 kHz)	153	6,553.6	25.6

This table is for 10MHz input clock. Scale appropriately for other clock rates:

Period (µs) = 512 x $2^{\text{Clock Select}}$ / (Input Clock in MHz)
Frequency (Hz) = 1,000,000 / Period (µs).

For 20MHz and 40MHz operation, the numbers should be scaled accordingly.

For pulsewidth long output, Table 3.8 gives the possible choices for pulsetrain repetition frequencies.

Table 3.8 Timer/Counter Pulsetrain Output

Clock Select	Frequency (Hz)	Period (ms)
0	76.3	13.1
1	38.1	26.2
2	19.1	52.4
3	9.54	105
4	4.77	210
5	2.38	419
6	1.19	839
7	0.60	1,678

This table is for 10MHz input clock. Scale appropriately for other clock rates:

Period (ms) = 131.072 x $2^{\text{Clock Select}}$ / (Input Clock in MHz)
Frequency (Hz) = 1,000 / Period (ms)

As with all CMOS devices, floating I/O pins can cause excessive current consumption. To avoid this, declare all unused I/O pins as bit output. Alternatively, unused I/O pins may be connected to + V_{DD} or GND.

4

Hardware Design Considerations

4.1 Introduction

This chapter covers the hardware design considerations for the use of the FT 3120 and FT 3150 Smart Transceivers. These design considerations include the interconnections to the Smart Transceiver and the FT-X1 transformer, PCB Layout guidelines, and EN 61000-4 EMC immunity testing.

4.2 Quick Start for Users Familiar With The FTT-10A Transceiver

For readers who are already familiar with the FTT-10A transceiver and its use with Neuron Chips, this section summarizes the differences between designing devices using the FTT-10A transceiver and designing devices using the FT 3120 and FT 3150 Smart Transceivers.

The FT 3120 and FT 3150 Smart Transceivers can be used along with the FT-X1 transformer in existing PCBs that have been designed for Neuron Chips and the FTT-10A transceiver. The FT 3120 and FT 3150 Smart Transceiver chips have the same footprints as the corresponding Neuron Chips. The FT-X1 transformer has the same footprint as the FTT-10A transceiver, and the FT-X1 transformer's pinout is compatible with the connections between the Neuron Chip and the FTT-10A transformer. Refer to the *FT 3120 and FT 3150 Smart Transceiver* datasheet for more detailed information on these pinouts and footprints.

If the FT 3120 and FT 3150 Smart Transceivers and the FT-X1 transformer are substituted for the Neuron Chip and the FTT-10A transceiver on an existing device design, the device should perform as it has in the past, with the same levels of transient immunity, with improved magnetic field noise immunity, and with improved common-mode network noise immunity (as tested per EN 61000-4-6). With a small component substitution and the addition of the two small capacitors C5 and C6, the common-mode network noise immunity can be further improved over FTT-10A transceiver-based devices. In Figure 4.1, capacitors C5 and C6 are added from T1 and T2 to ground to raise the EN61000-4-6 common mode noise immunity to Level 3, and there is a 470V metal-oxide varistor (MOV) VR1 in place of the 1000pF, 2kV capacitor that was used with the FTT-10A transceiver (see capacitor "C2" in Figure 2.1 and Figure 2.2 in the *FTT-10A Free Topology Transceiver Users's Guide*). Also, since the 470V MOV clamps network ESD transients before any spark gaps could fire, the spark gaps on Net1 and Net2 are no longer needed as they were with FTT-10A transceiver-based devices. Since spark gaps are no longer needed, the DSP-301 spark gap component-based ESD protection circuit (shown in Figure 2.2 in the *FTT-10A Free Topology Transceiver Users's Guide*) has been eliminated.

When using BAV-99-equivalent diodes for the differential clamp diodes D3-D6 in Figure 4.1, the device should pass EN61000-4-5 Surge testing to Level 3 (2kV), just as the old FTT-10A transceiver-based devices did. However, by using the larger 1N4935-equivalent diodes listed in Table 4.1 for D3-D6, you now should be able to achieve a higher 6kV surge immunity level. This is a new feature that was not previously available. However, 6kV surge immunity is not generally needed in LONWORKS devices, so you should only use the larger 1N4935-equivalent differential clamp diodes if your application would benefit from the higher surge immunity level. If you have a device that is already based on the larger diodes of Figure 2.2 in the *FTT-10A Free Topology Transceiver Users's Guide*, you can keep using them with the FT 3120 and FT 3150 Smart Transceivers, and just change the 1000pF, 2kV capacitor to the 470V MOV.

There are several other factors to consider in addition to changing the 1000pF, 2kV capacitor to the 470V MOV when migrating to the FT 3120 and FT 3150 Smart Transceivers. The Neuron Chip's RXD and TXD digital signal pins are now the T1 and T2 transformer analog lines between the Smart Transceiver and the FT-X1 transformer. Since these lines are now used for analog connections instead of digital connections, care should be taken in PCB layouts to keep these lines close together and away from noisy digital lines. Devices that followed the PCB layout guidelines in the *FTT-10A Free Topology Transceiver Users's Guide* will already have these two traces fairly well isolated from other signals. Since the clock line is no longer needed at the FT-X1 transformer position, the clock trace is not shown in the PCB layout figures later in this chapter. V_{CC} is still needed in the area of the FT-X1 transformer for use with the T1-T2 ESD clamp diodes D1-D2, as shown in Figure 4.3 later in this chapter.

The T1 and T2 signals are each brought out of the FT-X1 transformer at two different sets of pins. The T1 and T2 connections between the FT-X1 transformer and the Smart Transceivers can be made via either set of pins on the FT-X1 transformer, but the connections shown in the PCB layout figures later in this chapter generally give the best ESD transient immunity. T1 and T2 were used in FTT-10A transceiver-based designs for ESD clamping only, but now they are also connected directly to the FT 3120 and FT 3150 Smart Transceivers via the old RXD and TXD signals of the Neuron Chip.

Since the T1 and T2 lines are analog connections between the Smart Transceiver and the FT-X1 transformer, no other parallel connections should be made to these lines (other than the C5 and C6 capacitors shown in Figure 4.1), and no components should be placed in series between the Smart Transceiver and the FT-X1 transformer. In particular, no RXD/TXD sensing circuitry should be attached to these lines. There is now an explicit RXD/TXD signal available on the Smart Transceiver's COMM_ACTIVE pin which can be used for the purpose of driving RXD and TXD LEDs, if desired. See the following section for more information about the COMM_ACTIVE LED drive circuit.

4.3 Interface Between FT 3120 / FT 3150 Smart Transceiver and the Network

The preferred interconnection between the FT-X1 transformer, a Smart Transceiver and the associated transient protection circuitry is shown in Figure 4.1.

Figure 4.1 is not a complete schematic, since it does not include the clock, reset, and power supply bypass circuits for the Smart Transceivers. Refer to the *FT 3120 and FT 3150 Smart Transceiver Datasheet* for this additional information.

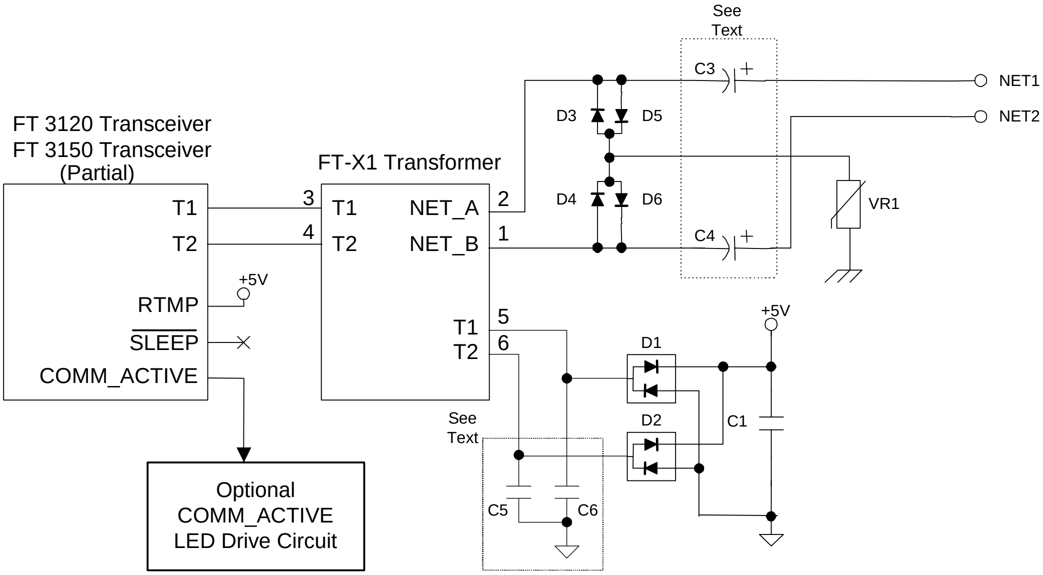


Figure 4.1 FT 3120 / FT 3150 Smart Transceiver and FT-X1 Interconnections with Transient Protection Circuitry

Table 4.1 FT 3120 / FT 3150 Smart Transceiver External Components

Name	Value	Comments
C1	0.1µF for +5VDC decoupling	V_{CC} decoupling capacitor for ESD protection diodes D1-D2
VR1	470V MOV, 5mm, 40pF (typ.)	Panasonic ERZV05D471, Digi-Key P7186-ND or equivalent.
C3, C4	22µF, ≥50V, polar	DC blocking capacitors; see text
C5, C6	56pF, 5%, NPO or COG	To enable EN61000-4-6 Level 3
D1, D2	BAV99, 1N4148-equivalent	ESD Transient clamping diodes
D3, D4, D5, D6		Differential network clamping diodes:
	BAV99, 1N4148-equivalent	For up to 2kV Surge Protection
	1N4934, 1N4935, FR1D, RS1D, RS1DB	For up to 6kV Surge Protection

In Figure 4.1, capacitors C3 and C4 are used to provide DC voltage isolation for the FT 3120 and FT 3150 Smart Transceiver when it is used on a link power network or in the event of a DC power fault on the network wires. The capacitors are required to meet LONMARK interoperability guidelines for the TP/FT-10 channel. These capacitors are not needed on devices that will be connected exclusively to non-link power networks and do not require protection against DC faults. Two polar capacitors are used to protect against the application of a DC voltage of either polarity, while providing a total capacitance of 11µF. Alternatively, a single non-polar capacitor of 10µF may be used in either of the two legs which connect to the network. The initial tolerance of the capacitor should be ±20% or less, and degradation due to aging and temperature effects should not exceed 20% of the initial minimum value

Capacitors C5 and C6 are required on all new designs. They ensure that the FT 3120 and FT 3150 Smart Transceivers support EN61000-4-6 Level 3. Note that unlike the FTT-10A transceiver, the common mode noise immunity of the FT 3120 and FT 3150 Smart Transceivers is not significantly improved by the addition of the common mode choke specified in the *LONWORKS FTT-10A Free Topology Transceiver User's Guide*.

Figure 4.2 shows the optional COMM_ACTIVE LED drive circuit block referred to in Figure 4.1 on the previous page.

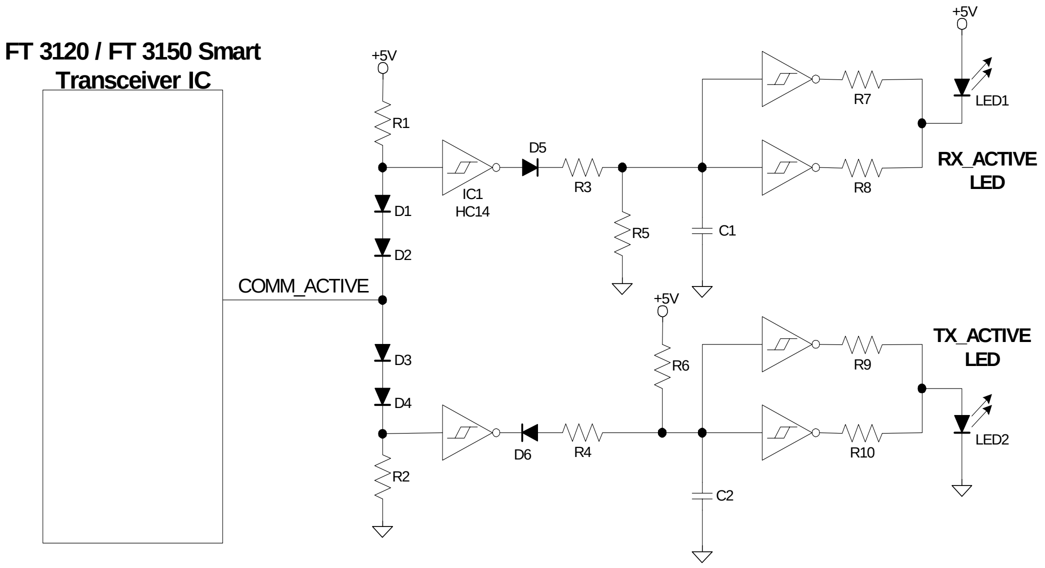


Figure 4.2 Optional COMM_Active LED Drive Circuit

The COMM_ACTIVE LED drive circuit shown in Figure 4.2 is for interfacing the FT 3120 or FT 3150 Smart Transceiver COMM_ACTIVE pin to LEDs. These TX_ACTIVE and RX_ACTIVE LEDs can be used to provide visual indication of physical transceiver activity.

The COMM_ACTIVE pin is normally in a high-impedance state, but it is driven high for the duration of a transmitted packet, and it is driven low for the duration of a received packet. The COMM_ACTIVE pin is driven low during the reception of a packet from the network, regardless of whether that packet is addressed to the device, and regardless of whether the packet has a valid CRC (which cannot be checked until the entire packet is received). Electrical and mechanical specifications for the Smart Transceivers and the pin number for the COMM_ACTIVE signal are documented in the *FT 3120 and FT 3150 Smart Transceiver Datasheet.*

Table 4.2 lists the component values for a COMM_Active circuit.

Table 4.2 COMM_Active Circuit Component Values

Component Name	Value
R1, R2	10kΩ
R3, R4	1kΩ
R5, R6	220kΩ
R7, R8, R9, R10	470Ω
C1, C2	0.1uF
D1, D2, D3, D4, D5, D6	1N4148-equivalent or BAV-99-equivalent
IC1	74HC14

Note: All resistors are ≥ 1/10 Watt. Resistor tolerances can be 1% or 5%.

4.4 PC Board Layout Guidelines

An example of a PC board layout is shown in Figure 4.3. The layout uses surface mount (SMT) components on the top side of the PCB. The scale of the figures is approximately 4X, but they are not intended for use as finished PCB artwork. Variations on this suggested PCB layout are possible as long as the general principles discussed later in this chapter are followed. Through-hole capacitors and diodes can be used, but SMT components will generally be superior because of their lower series inductance.

Electrostatic discharge (ESD) and electromagnetic interference (EMI) are two of the most important design considerations when laying out the PCB for a device. These topics are discussed in general terms at the end of this chapter. The specifics relating to PCB layout issues are discussed in this section.

Tolerance of ESD and other types of network transients requires good layout of the power, ground and other device circuitry. In general, an ESD discharge current will return to earth ground or other nearby metal structures. The device's ground scheme must be able to pass this ESD current between the network connection and the device's external ground connection without generating significant voltage gradients across the device's PCB. The low-inductance Star ground scheme illustrated in Figure 4.4 and discussed below accomplishes this task.

1. **Star Ground Configuration:** The distribution of functional circuit blocks on the PCB should be in the form of a star, with the power connector, network connector and any chassis ground connection all located as close as practical to the center of the star. This star ground distribution is illustrated in Figure 4.4. The goal of star ground distribution is to conduct transients out of the device with minimal disruption to other function blocks. If the device has a metal chassis, then ESD and other transients will generally return to that chassis via the star ground center point. If the device's logic ground is connected to this chassis ground, then connection should only be made at this single point in the center of the star ground. Logic ground and chassis ground are shown connected in Figures 4.3 and 4.4. If a device is housed in a plastic enclosure and is powered with an isolating transformer, then there may not be any explicit earth ground or chassis ground available. In this case, it is still important for the network connector and power supply connector to be located near the center of the star.
2. **ESD Keepout Area:** The PC board layout should be designed so that substantial ESD hits from the network will discharge directly to the star ground center point. This is accomplished by the placement of the 470V MOV VR1, near the network connector and near the center of the star ground. This shunts the majority of the network ESD hit energy directly to the star center, which helps to limit the transient current that passes through the FT-X1 transformer. The keep-out area noted in Figure 4.3 is designed to eliminate unintended ESD discharge paths.
3. **D1/D2 Clamp Diodes:** The D1 and D2 diodes clamp the Smart Transceiver side of the FT-X1 transformer between V_{CC} and ground. The V_{CC} and ground connections between D1, D2, and the FT-X1 transformer must be made using the low inductance technique shown in Figure 4.3. This ensures that the secondary transient energy (remaining after the primary discharge through the MOV VR1) does not disrupt the Smart Transceiver. The V_{CC} and ground connections of diodes D1 and D2 are designed to return transient currents to the star ground center point.
4. **D3-D6 Clamp Diodes:** The D3 through D6 diodes in Figure 4.1 (shown as D3 and D4 in Figure 4.3) clamp the network side of the FT-X1 transformer to ground through the MOV VR1 during ESD and surge transients. The connections between D3-D6, and VR1 must be made using the low inductance technique shown in Figure 4.3. This ensures that secondary transient energy remaining after the primary discharge through the MOV does not disrupt the Smart Transceiver. The connection of VR1 is designed to return transient currents to the star ground center point.
5. **Ground Planes:** As ground is routed from the center of the star out to the function blocks on the board, planes or very wide traces should be used to lower the inductance (and therefore the impedance) of the ground distribution system.
6. **+5V Power Distribution & Decoupling:** In general, V_{CC} should be distributed through low inductance traces and planes in the same manner as ground. All of the ground pins on the FT 3120 and FT 3150 Smart Transceivers should be connected with either a ground plane (for PCBs with at least 4 layers) or a ground pad directly underneath the Smart Transceiver chip on the component side of the board (for PCBs with 2 layers). At least three 0.1µF SMT decoupling caps are recommended around the Smart Transceiver on the component side of the PCB. See

figures C.9, C.10 and C.11 for more details on grounding and V_{CC} decoupling for FT 3120 and FT 3150 Smart Transceivers. The ESD decoupling capacitor C1 should be placed immediately next to the FT-X1 transformer and diodes D1-D2, as shown in Figure 4.3.

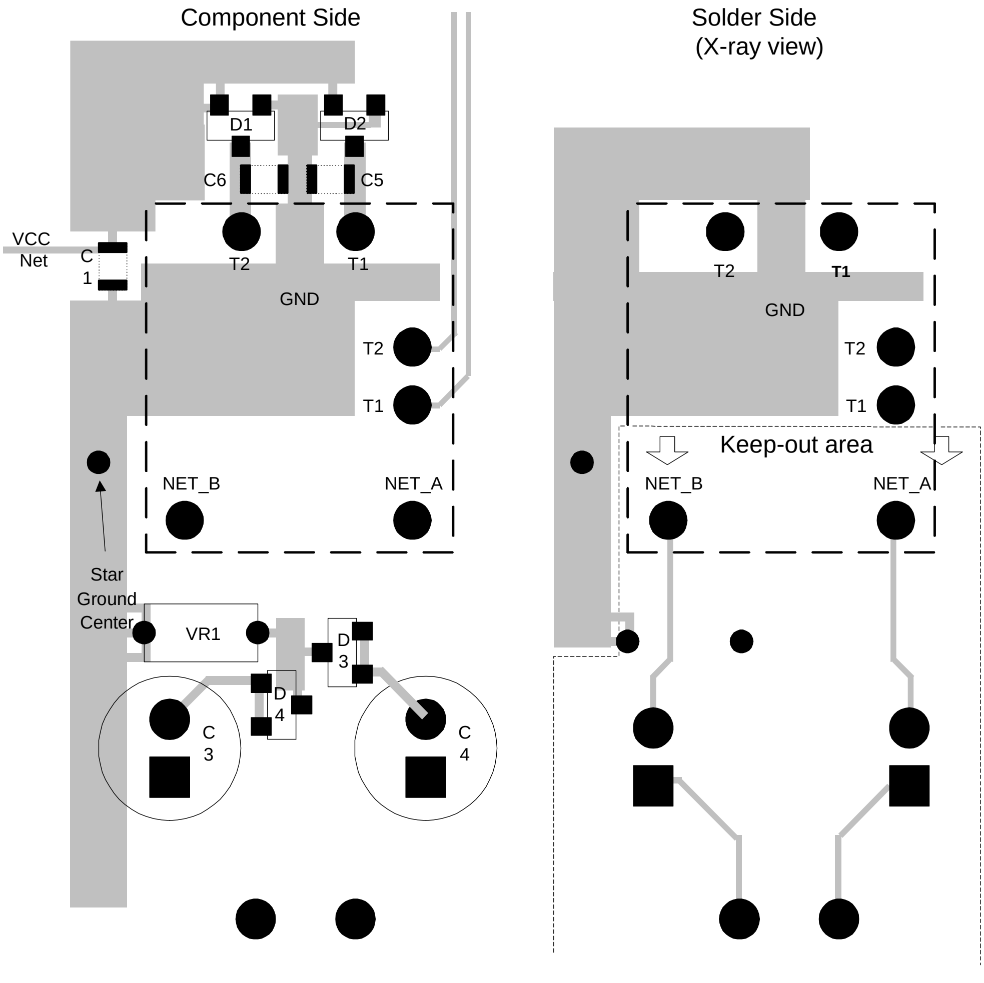


Figure 4.3 Smart Transceiver PCB Layout

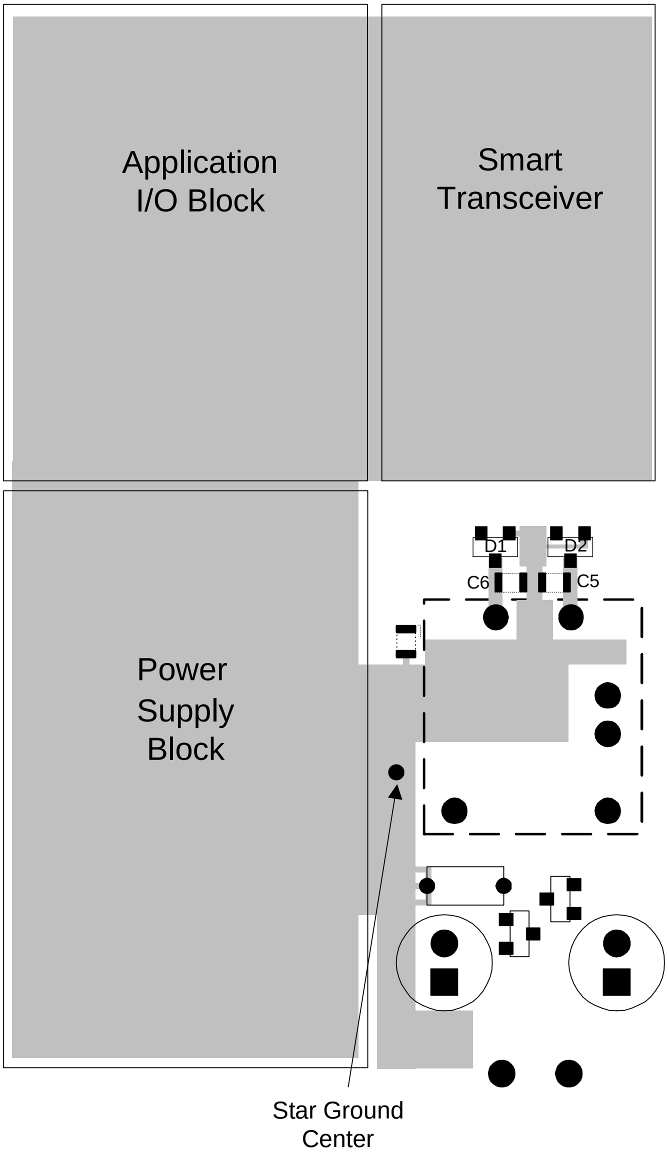


Figure 4.4 Star Ground Design

4.5 EMI Design Issues

The high-speed digital signals associated with microcontroller designs can generate unintentional electromagnetic interference (EMI). High-speed voltage transitions generate RF currents that can cause electromagnetic radiation from a product if a length of wire or piece of metal serves as an antenna.

Products that use a Smart Transceiver will generally need to demonstrate compliance with EMI limits enforced by various regulatory agencies. In the USA, the FCC requires that unintentional radiators comply with Part 15 level “A” for industrial products, and level “B” for products that can be used in residential environments. Similar regulations are imposed in most countries throughout the world.

Careful PCB layout is important to ensure that a Smart Transceiver based-device will achieve the desired level of EMC. A typical Smart Transceiver based-device will have several digital signals switching in the 1-40MHz range. These signals will generate voltage noise near the signal traces, and will also generate current noise in the signal traces and power supply traces. The goal of good device design is to keep this voltage and current noise from coupling out of the product's package.

It is very important to minimize the leakage capacitance from circuit traces in the device to any external pieces of metal near the device, because this capacitance provides a path for the digital noise to couple out of the product's package. Figure 4.5 shows the leakage capacitances to earth ground from a device's logic ground ($C_{leak,GND}$) and from a digital signal line in the device ($C_{leak,SIGNAL}$). If the Smart Transceiver based-device is housed inside of a metal chassis, then that metal chassis will probably have the largest leakage capacitance to other nearby pieces of metal. If the device is housed inside of a plastic package, then PCB ground guarding must be used to minimize $C_{leak,SIGNAL}$. Effective guarding of digital traces with logic ground reduces $C_{leak,SIGNAL}$ significantly, which reduces the level of common-mode RF currents driven onto the network cable.

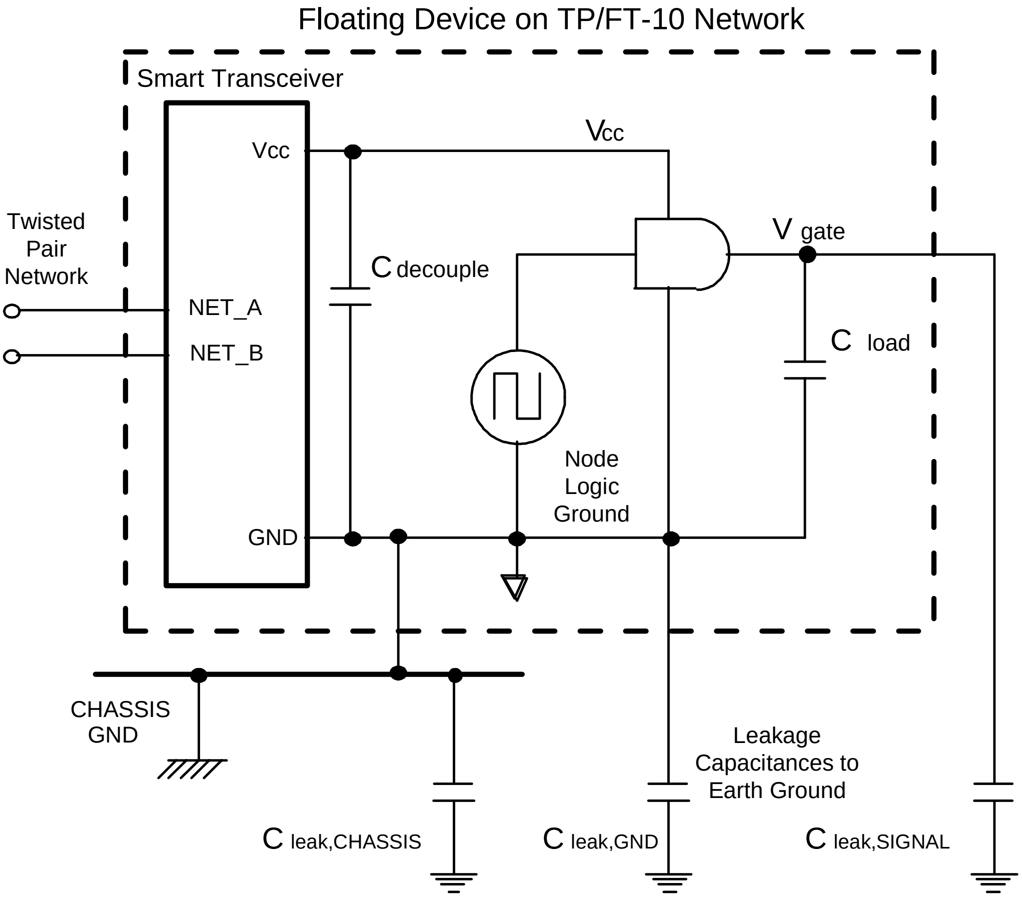


Figure 4.5 Parasitic Leakage Capacitances to Earth Ground

When a device is mounted near a piece of metal, especially metal that is earth grounded, any leakage capacitance from fast signal lines to that external metal will provide a path for RF currents to flow. When V_{gate} is pulled down to logic ground, the voltage of logic ground with respect to earth ground will increase slightly. When V_{gate} pulls up to V_{CC}, logic ground will be pushed down slightly with respect to earth ground. As $C_{leak,SIGNAL}$ increases, a larger current flows during V_{gate} transitions, and more common-mode RF current couples into the network twisted pair. This common-mode RF current can generate EMI in the 30-500MHz frequency range in excess of

FCC/CISPR “B” levels, even when $C_{leak,SIGNAL}$ from a clock line to earth ground is less than 1pF. Guarding of clock lines is essential for meeting Level “B” limits.

From this discussion, it should be apparent that minimizing $C_{leak,SIGNAL}$ is very important. By using 0.1µF or 0.01µF decoupling capacitors at each digital IC power pin, V_{CC} and logic ground noise can be reduced. Logic ground can then be used as a ground shield for other noisy digital signals and clock lines.

Since the FT 3150 Smart Transceiver based-device has an external memory interface bus, there are many more traces that need to be guarded by logic ground in a FT 3150 Smart Transceiver based-device. In addition, the V_{CC} noise generated by the memory interface and external ROM/RAM components requires more V_{CC} decoupling, and may require a four-layer PCB to maintain an RF-quiet V_{CC} and logic ground.

Some Smart Transceiver based-devices with fast digital circuitry, such as MIP-based devices, DSP engines, and large memory arrays, may require extra RF attenuation between the FT-X1 transformer and the twisted pair network in order to meet FCC/CISPR level “A” or “B”. This extra attenuation can be provided by a ferrite bead (muRata BLM11A601 or equivalent) in series with each network line adjacent to the network connector. Each of these ferrite beads must have an inductance of no more than 30µH. Alternately, a common mode EMI choke (muRata PLT1R53C or equivalent) could be used in place of the two ferrite beads. If a common-mode choke is placed in series with the network connection, it must have less than 40pF of differential capacitance. If the FTT-10A Transceiver’s Network Isolation Choke was used in the past between the transceiver and the network for EMI attenuation on a particular device, then that choke can still be left in place with the Smart Transceiver and the FT-X1 transformer. However, the Smart Transceiver and FT-X1 transformer generally do not gain any benefit from the common-mode choke for EN 61000-4-6 compliance.

Some amount of filtering may also be required on an Smart Transceiver device's power supply input, depending on the level of noise generated by the application circuitry. This is best accomplished by placing ferrite chokes in series with the power input traces adjacent to the power connector. A typical power supply circuit illustrating the placement of these ferrite chokes is shown in Figure 4.6.

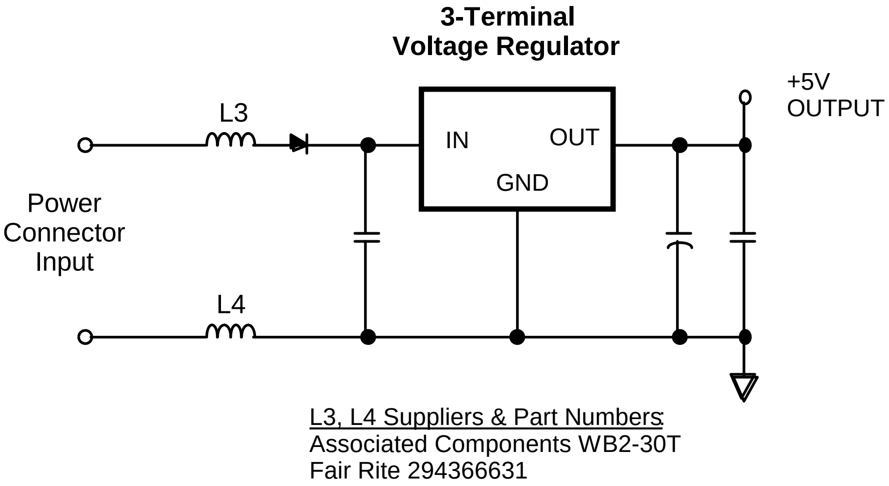


Figure 4.6 Illustration of Power Supply Input Filtering Using Ferrite Chokes

In summary, the following general rules and guidelines apply:

- The faster the Smart Transceiver clock speed, the higher the level of EMI.
- Better V_{CC} decoupling quiets RF noise at the sources (the digital ICs), which lowers radiated EMI.
- The FT 3120 Smart Transceiver will generate less EMI than the FT 3150 Smart Transceiver because the FT 3120 Smart Transceiver has no external memory interface lines.
- A four-layer PCB will generate less EMI than a two-layer PCB since the extra layers facilitate better V_{CC} decoupling and more effective logic ground guarding.
- A two-layer PCB Smart Transceiver based-device should be able to meet FCC/CISPR level "B" EMC if good decoupling and ground guarding are used.
- Ferrite beads in series with the network traces at the network connector, and ferrite chokes in series with the power input traces at the power connector, can be used to help meet EMC requirements for devices that have noisy application circuitry or special circuit requirements.

Early testing of prototype circuits at an outdoor EMI range should be used to determine the effectiveness of these EMC techniques in a particular application.

4.6 ESD Design Issues

Electrostatic discharge (ESD) is encountered frequently in industrial and commercial use of electronic systems. In addition, the European Community has adopted requirements for ESD testing.

Reliable system designs must consider the effects of ESD and take steps to protect sensitive components. Static discharges occur frequently in low-humidity environments when operators touch electronic equipment. Keyboards, connectors, and enclosures may provide paths for static discharges to reach ESD sensitive components such as the Smart Transceiver. This section describes the issues involved with designing ESD immunity into Smart Transceiver-based products.

There are two general methods that are used to ESD harden products. The first is to seal the product to prevent static discharges from reaching the sensitive circuits inside the package. The second method involves designing the grounding of a product so that ESD hits to user-accessible metal parts can be shunted around any sensitive circuitry.

Since the network connector is user-accessible, it is not possible to seal Smart Transceiver based-devices completely. However, the product's package should be designed to minimize the possibility of ESD hits arcing into the device's circuit board. If the product's package is made of plastic, then the PCB should be supported in the package so that unprotected circuitry on the PCB is not adjacent to any seams in the package. The PCB should not touch the plastic of an enclosure near a seam, since a static discharge can creep along the surface of the plastic, through the seam, and arc onto the PCB.

Once an ESD hit has arced to the product, the current from the discharge will flow through all possible paths back to earth ground. The grounding of the PCB and the protection of user-accessible circuitry must allow these ESD return currents to flow back to earth ground without disrupting normal circuit operation of the Smart Transceiver or other device circuitry. Generally, this means that the ESD currents should be shunted to the center of a star ground configuration (see Figure 4.4) and then out to the product's chassis or earth ground connection. If the device is floating with respect to earth ground, the ESD current will return capacitively to earth via the network wire, the power supply wires, and the PCB ground plane.

Designers of Smart Transceiver-based devices should follow the PCB layout guidelines presented earlier in this chapter. In addition, external clamping of user-accessible circuitry is required to shunt ESD currents from that circuitry into the center of the star ground on the PCB. For example, if the Smart Transceiver is scanning a keypad with some of its I/O lines, then the I/O lines to that keypad will need to be diode-clamped as shown in Figure 4.7. If a negative ESD hit discharges into the keypad, then the diode clamps to ground shunt the ESD current into the ground plane. If a positive ESD hit discharges into the keypad, then the V_{CC} diodes shunt the current to the ground plane via a 0.1µF decoupling capacitor that is placed directly adjacent to the clamp diodes. The keypad connector, diodes and decoupling capacitor should all be located close to the center of the star ground so that the ESD current does not pass through sensitive circuitry on its way out of the PCB.

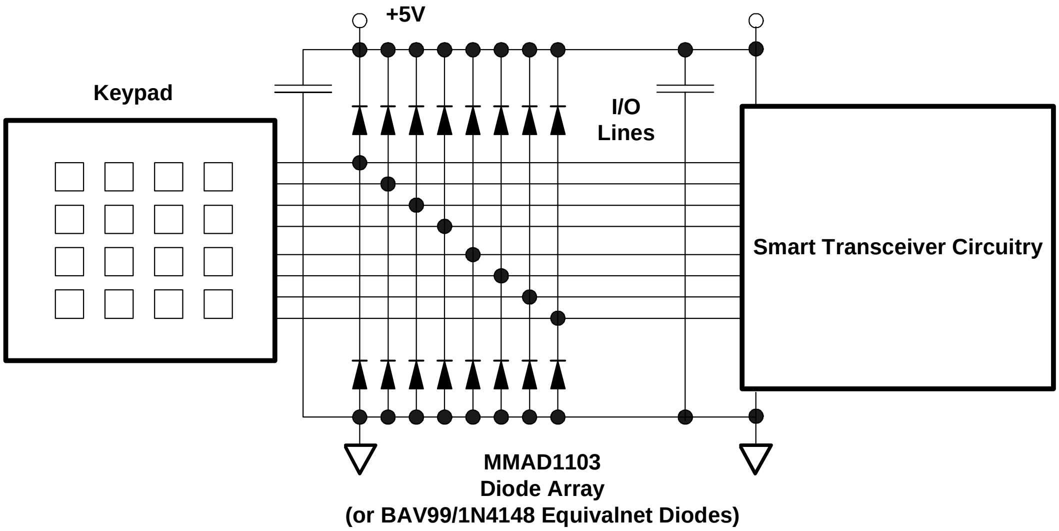

+5V
Keypad
I/O
Lines
Smart Transceiver Circuitry
MMAD1103
Diode Array
(or BAV99/1N4148 Equivalnet Diodes)

Figure 4.7 Illustration of I/O Line ESD Clamps

4.7 Building Entrance Protection

Echelon recommends using shielded twisted pair wire for all networks, or network portions, that are run outside of buildings. The shield should be connected to earth at each building entry point via a data lightning/surge arrester in order to conduct excessive surges or lightning strike energy to ground and prevent their entry inside the building via the network. Data line lightning/surge arresters should also be used at each building entrance and connected to the network data lines. Therefore, two additional arresters, one for each conductor of the twisted pair, are required, and the arresters must be of the gas discharge type to minimize any corruption of the network data traffic. MOV and TVS protection devices must not be used directly on the network data lines since the intrinsically higher capacitance of these devices will corrupt, or even prevent, network traffic between devices. This prohibition does not apply to the specific case of the use of the MOV VR1 as discussed at the beginning of this chapter, and shown in Figure 4.1 and Figure 4.3.

Gas discharge technology is typically used for data line lightning/surge arresters due to its very low capacitance (typically < 5pF) and rapid response. Low capacitance gas discharge arresters do not have any impact on the maximum number of devices possible on a network segment. For each 300 picofarads of protection circuitry capacitance connected to the channel, the TP/FT-10 device count must be reduced by one. For example, if two lightning/surge arresters, each presenting a 300 picofarad load, are used in the network, then the maximum number of devices allowed on that physical segment will be 64 minus 2 (62 devices). When using link power networks, the capacitance limit per link power device is 150 picofarads; if the lightning/surge arrester set presents 150 picofarads to the network cable at both entrances, the maximum number of link power devices is reduced from 128 to 126 devices. The above limits apply to any contiguous network segment. Repeaters and routers physically isolate their two segments so the limits stated apply to each segment. Also, due to the minimal amount of “leakage capacitance” to ground allowed in a link power network, the two arresters at each entrance in a link power network segment must be balanced to within 10 picofarads.

Figure 4.8 depicts twisted pair networks running in outdoor locations.

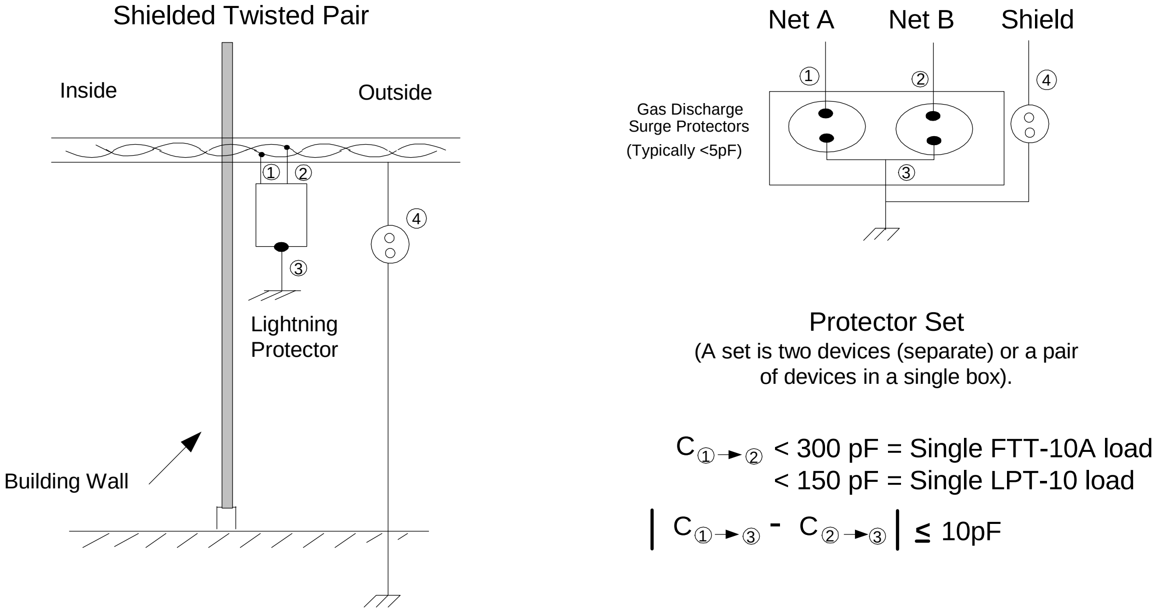


Figure 4.8 Network and Shield Lightning Protection

It is important that **every** FT 3120 and FT 3150 Smart Transceiver-based device on the network segment, whether located indoors or outdoors, has the protection circuitry shown in Figure 4.1. This is *in addition* to any building entrance protection devices present on the network.

4.8 EN 61000-4 Electromagnetic Compatibility (EMC) Testing

Echelon has tested the FT 3120 and FT 3150 Smart Transceivers operating in typical two-layer application boards to verify that they comply with the five applicable EN 61000-4 test requirements (formerly known as IEC 1000-4 tests). Provided that a device's PCB is designed following the guidelines in this chapter, the Smart Transceivers should pass the EN 61000-4 tests to the levels described in Table 4.3.

Table 4.3 EN 61000-4 Test Immunity Levels

EN Test	Description	Smart Transceiver Immunity Level
EN 61000-4-2	ESD Immunity	Level 4
EN 61000-4-3	Radiated Susceptibility	Level 3
EN 61000-4-4	Burst Immunity	Level 4
EN 61000-4-5	Surge Immunity	Level 3 (2kV) with D3-D6 = BAV-99-equiv. diodes in Figure 4.1 Level "X" (6kV) with D3-D6 = 1N4935-equiv. diodes in Figure 4.1
EN 61000-4-6	Conducted RF Immunity	Level 3 (C5 =C6 = 56pF caps required)

EN 61000-4-2 ESD testing is performed on a metal test table using an ESD transient generator. Level 4 testing involves injecting up to ±8kV contact discharges and up to ±15kV air discharges into the product under test. Depending upon the product design, discharges may be injected at the network connector, power connector and other user-accessible areas. During the test, proper device operation should continue with occasional loss of a packet due to the ESD hits.

EN 61000-4-3 RF Susceptibility testing is generally performed in an RF-shielded anechoic chamber. The product under test is placed on a non-conducting table in the chamber, and antennas are used to subject the product to intense radio frequency fields. Under the test, proper operation continues with the occasional loss of a packet. Level 2 testing is performed with a field of 3V/m, which is classified by the test standard as a 'moderate electromagnetic radiation environment.' Level 3 testing is performed with a field of 10V/m, which is classified by the standard as a 'severe electromagnetic radiation environment.'

EN 61000-4-4 Burst testing is performed on a non-conducting table, with 1 meter of the network cable clamped in a high-voltage burst generation apparatus. Under the test, proper operation continues with occasional loss of a packet. There are three bursts injected onto the network cable each second. Level 3 testing is performed with ±1kV bursts, which are classified by the test standard as representative of a 'typical industrial environment.' Level 4 testing is performed with ±2kV bursts, which are representative of a 'severe industrial environment.'

EN 61000-4-5 Surge testing is performed on a non-conducting table using specialized surge generation equipment. The surges are injected directly into the network wiring via a coupling circuit. See Figure 10 of EN 61000-4-5 (formerly Figure 11 of IEC 801-5). Under the test, proper operation continues with the occasional loss of a packet.

Level 2 testing is performed with up to ±1kV surges, and Level 3 testing is performed with up to ±2kV surges. Level "X" testing is performed at a user-defined surge voltage (6kV in the case of the Smart Transceivers).

For more information on levels and installation classes, see EN 61000-4-5. The applicable surge test levels and coupling mode specified by EN 61000-4-5 can be found in Table A.1 of that document as follows:

- Balanced circuits/lines;
- Coupling mode is line-to-ground, either polarity;
- Surge waveform is 1.2/50µs (8/20µs) combination wave for classes 1-4.

EN 61000-4-6 Conducted RF Immunity testing is performed on a metal test table using an RF signal generator, an RF power amplifier, a current injection clamp, and specialized "coupling-decoupling" network (CDN) devices. A typical test setup for use with LONWORKS devices is shown in Figure 4.9. The test equipment drives a large common-mode noise voltage onto the twisted pair cable that connects the Auxiliary Equipment (AE) and the Equipment Under Test (EUT). The AE must be able to continue communicating with the EUT during the test. The network error rate for this communication should generally be less than 1%, indicating a negligible loss of network functionality. During the test, the RF signal generator is set to an amplitude modulation (AM) depth of 80%, and the frequency is slowly swept from 150kHz to 80MHz. Level 2 testing, which represents a "light industrial environment," is performed with an injected common-mode voltage on the EUT's network cable of 3Vrms (15.3Vp-p including the 80% AM). Level 3 testing, which represents a "harsh industrial environment," is performed with an injected common-mode voltage on the EUT's network cable of 10Vrms (50.9Vp-p including the 80% AM).

The Current Injection method (sometimes called the Bulk Current Injection or BCI method) is the preferred test method for twisted pair networks. A current clamp is used to inject common mode noise onto the twisted pair communication cable, and both the AE and the EUT experience similar common mode noise at their network connections. Because of this, it is generally best if the AE can provide some visual indication of pass/fail during the test, rather than requiring a wired connection back to a computer to monitor pass/fail. Even when this wiring passes through a CDN, the RF noise present during the test may disrupt wired communication between the AE and an external control PC.

A typical setup for EN 61000-4-6 testing of a Smart Transceiver-based device and unshielded twisted pair (UTP) network wire is shown in Figure 4.9.

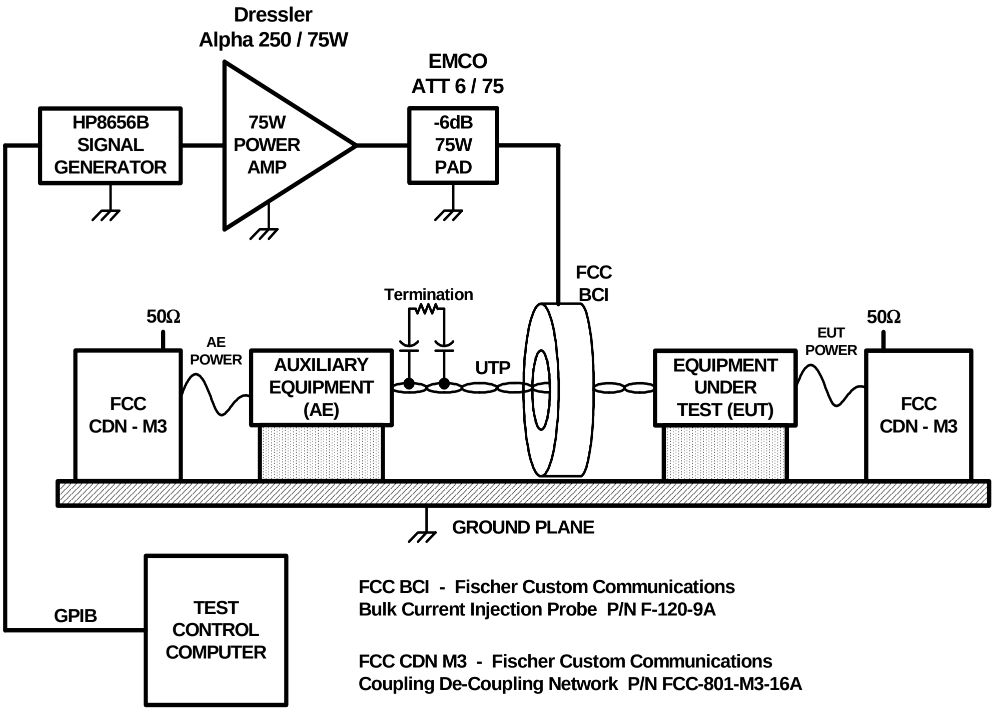
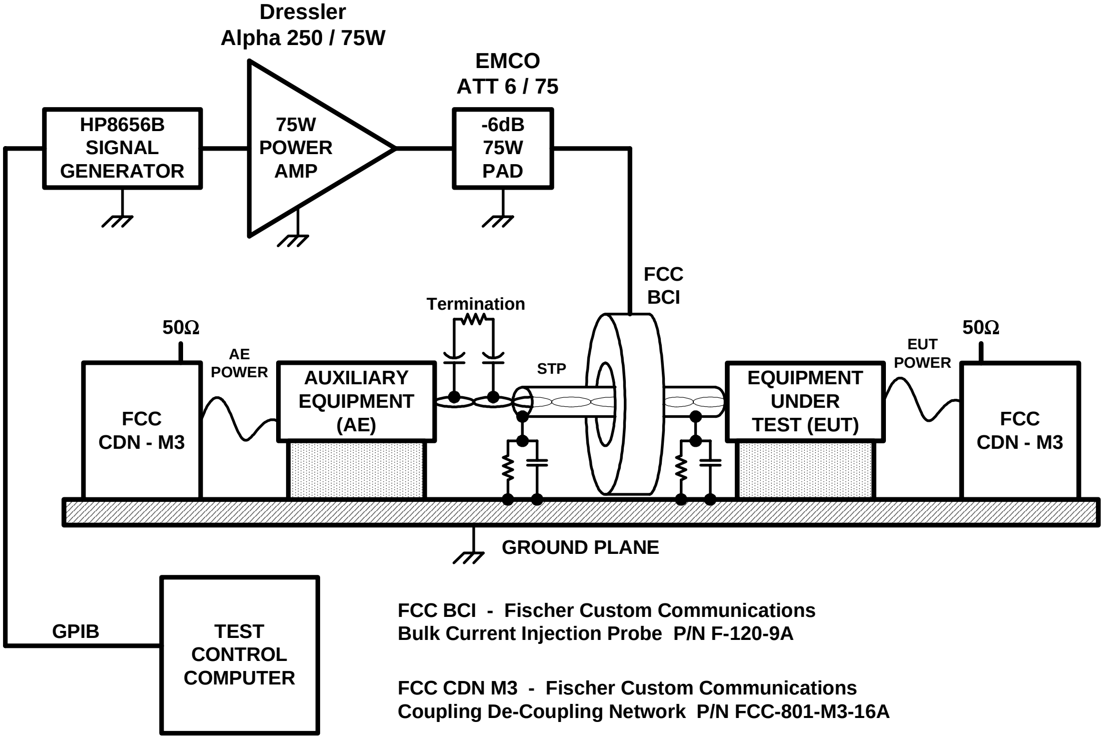


Figure 4.9 Typical EN 61000-4-6 Test Set-up for Unshielded Twisted Pair (UTP)

For the EN 61000-4-6 tests, the EUT is placed on a 10cm high, non-conducting support on top of the metal ground plane. If the EUT’s chassis is connected to earth ground in typical installations, it should be connected directly to the metal ground plane during the EN 61000-4-6 tests via a short wire. If the EUT is left floating in normal use, there should be no connection between the EUT and earth ground for the EN 61000-4-6 tests. The power connections for the AE and EUT should be routed through suitable decoupling devices, such as the non-driven M3 CDNs shown in the figure. During the network immunity tests, any I/O lines that come out of the AE or EUT should also pass through a decoupling network. The objective of the BCI current clamp in the figure is to drive the large common-mode noise signal into the EUT’s network cable. The M3 CDNs in the figure ensure that the power supply inputs to the AE and EUT are not an RF return path for the purposes of the EN 61000-4-6 test. See the EN 61000-4-6 test standard and related articles for more information about test setups and procedures.



FCC BCI - Fischer Custom Communications
Bulk Current Injection Probe P/N F-120-9A

FCC CDN M3 - Fischer Custom Communications
Coupling De-Coupling Network P/N FCC-801-M3-16A

Figure 4.10 Typical EN61000-4-6 Test Set-up for Shielded Twisted Pair (STP)

Figure 4.10 shows the changes in the EN 61000-4-6 test setup to accommodate shielded twisted pair (STP) networks. The BCI current clamp injects common mode noise onto the STP cable, and the cable shield should be connected to earth ground with a parallel resistor and capacitor as shown. The resistor is generally 470kΩ, 1/4W, 5%. The capacitor is generally 0.1uF, 10%, metal polyester, with a voltage rating of 100V or higher.

As indicated in Table 4.3, Smart Transceiver-based devices will generally pass the higher 10Vrms (50.9Vpp) test level with little or no interference with data communications.

Figure 4.11 shows a worst-case example of the EN 61000-4-6 CM noise immunity of FT 3120 and FT 3150 Smart Transceivers, and for comparison it also shows the typical immunity of the FTT-10A transceiver (without the FTT-10A Network Isolation Choke).

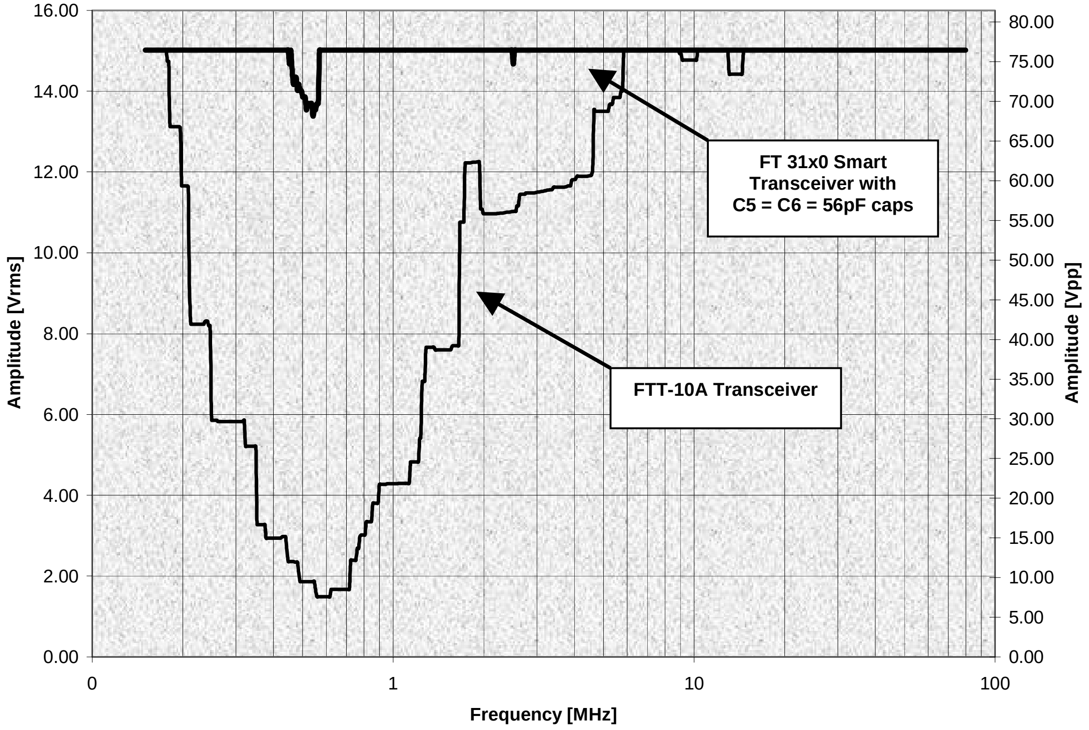


Figure 4.11 Smart Transceiver Common-Mode Noise Immunity

5

Network Cabling and Connections

5.1 Network Connection

The network connection (NET1 and NET2) is polarity insensitive. Therefore, either of the two twisted pair wires can be connected to either of these network connections.

5.2 Network Topology Overview

The TP/FT-10 network is designed to support free topology wiring, and will accommodate bus, star, loop, or any combination of these topologies. LPT-10 Transceiver-based devices and FT 3120 / FT 3150 Smart Transceiver-based devices can be located at any point along the network wiring. This capability simplifies system installation and makes it easy to add devices if the network need to be expanded. Figures 5.1 through 5.5 present five different network topologies. The actual termination circuit will vary by application.

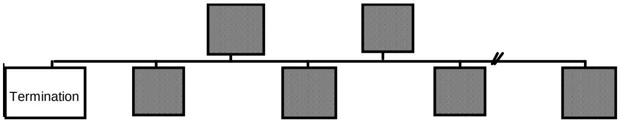


Figure 5.1 Singly Terminated Bus Topology

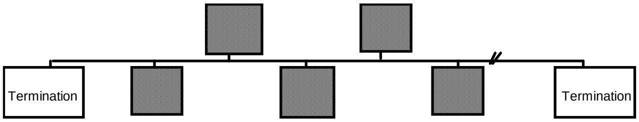


Figure 5.2 Doubly Terminated Bus Topology

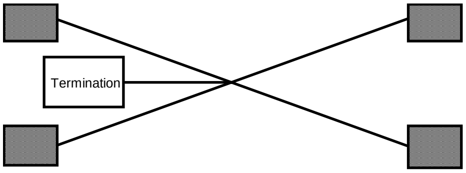


Figure 5.3 Star Topology

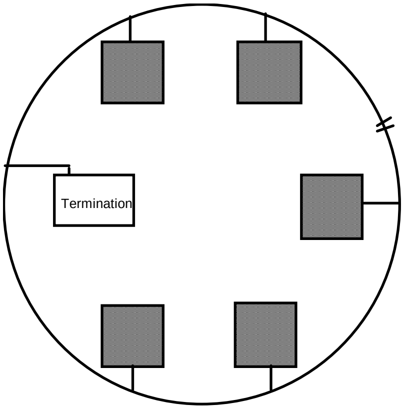


Figure 5.4 Loop Topology

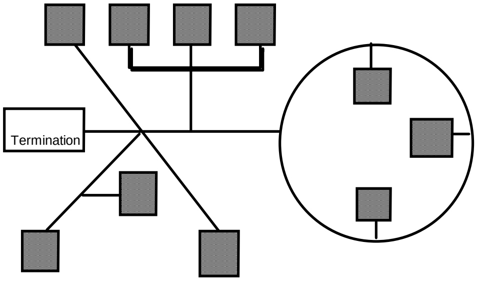


Figure 5.5 Mixed Topology

In the event that the limits for the number of transceivers or total wire distance are exceeded, then one FTT physical layer repeater (PLR) can be added to interconnect two segments and double the overall system capability. FTT-10A Transceivers are used to make physical layer repeaters. See the *FTT-10A Free Topology Transceiver User's Guide* for more info on PLRs.

5.3 System Performance and Cable Selection

TP/FT-10 channels must meet the *System Specifications* and *Transmission Specifications* as outlined below. When creating documentation and installation procedures for TP/FT-10 channels, be sure to include the TP/FT-10 system specifications and, based upon the cable used, the appropriate transmission specification listed below. Incorporating these specifications will ensure a smoother installation and provide a resource for the installer who must troubleshoot the installation.

Echelon has qualified a variety of cables for use with TP/FT-10 channels. Based on the cost, performance, and availability of these different cable types, system designers can choose the most appropriate cable for their application. Echelon has qualified five "generic" cable types, as follows:

- A generic 16AWG (1.3mm diameter) cable (similar to Belden 85102);
- NEMA Level 4 cable (this cable is ***not*** equivalent to TIA Category IV cable); and
- TIA Category 5 cable.

The electrical specifications for these cables can be found in Appendix B. A list of cable vendors can be found in the *Junction Box and Wiring Guidelines for Twisted Pair LONWORKS Networks* engineering bulletin (005-0023-01). This bulletin can be found on Echelon's web site at www.echelon.com. These cables have been qualified by Echelon in a generic form, and are available from vendors in a number of variations, including shielded, unshielded, plenum, and non-plenum jacketing. Additionally, Echelon has qualified two 16AWG (1.3mm) Belden cables, and one cable used in specific applications in the European market (JY(St)Y).

Note: The following specifications are for one network segment. Multiple segments may be combined using FTT-10A based physical layer repeaters or LonPoint routers to increase the number of devices and distance as described in the *FTT-10A Free Topology Transceiver User's Guide.*

5.3.1 System Specifications

- Up to 64 FT-X1 Transformers and FT 31X0 Smart Transceivers are allowed per network segment.
- LPT-10 transceivers may be used on network segments with FTT-10A transceivers and FT 3120 / FT 3150 Smart Transceivers, but are subject to additional constraints, particularly on distance. See the *LONWORKS LPT-10 Link Power Transceiver User's Guide* for more information.
- The average temperature of the wire must not exceed +55°C, although individual segments of wire may be as hot as +85°C.
- As a general rule, the TP/FT-10 channel communication cables should be separated from high voltage power cables. Follow local electrical codes with regard to cable placement.

5.3.2 Transmission Specifications

Table 5.1 Doubly-Terminated Bus Topology Specifications

Cable	**Maximum bus length (Meters)**
Belden 85102	2700
Belden 8471	2700
Level IV, 22AWG	1400
JY(St) Y 2x2x0.8	900
TIA Category 5	900

A doubly-terminated bus may have stubs of up to 3 meters from the bus to each device.

Table 5.2 Free Topology Specifications

Cable	**Maximum device-to-device distance**	**Maximum total Wire length (Meters)**
Belden 85102	500	500
Belden 8471	400	500
Level IV, 22AWG	400	500
JY(St) Y 2x2x0.8	320	500
TIA Category 5	250	450

The free topology transmission specification includes two components that must both be met for proper system operation. The distance from each transceiver to all other transceivers and to the termination (including the LPI-10 termination, if used) must not exceed the *maximum device-to-device distance*. If multiple paths exist, e.g., a loop topology, then the longest path should be used for calculations. The *maximum total wire length* is the total length of wire within a segment.

5.4 Cable Termination and Shield Grounding: Terminators, Termination Procedures, and Grounding Circuits

TP/FT-10 network segments require termination for proper data transmission performance. The type of terminator varies depending on whether shielded or unshielded cable is used. Free topology and Bus topology networks also differ in their termination requirements. The following sections describe the various terminators and termination procedures.

5.4.1 Free Topology Network Segment

In a free topology segment, only one termination is required and may be placed anywhere on the free topology segment. There are two choices for the termination:

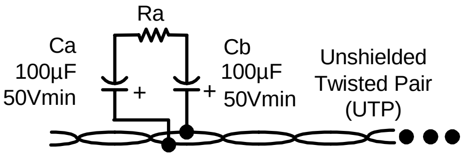
1. RC network (Figure 5.6), with Ra = 52.3Ω±1%, 1/8W
2. LPI-10 Link Power Interface, with jumper at “1 CPLR” setting.

5.4.2 Doubly Terminated Bus Topology Segment

In a doubly terminated bus topology, two terminations are required, one at each end of the bus. There are two choices for each termination:

3. RC network (Figure 5.6), with Ra = 105Ω±1%, 1/8W
4. LPI-10 Link Power Interface, with jumper at “2 CPLR” setting.

Only one LPI-10 interface is supported per segment. The LPI-10 contains the two required terminators. The other terminator must be an RC-type (see figure below).



Notes:

- Ca and Cb are typically aluminum-electrolytic type for improved longevity in the presence of ESD – observe polarity.
- Ca and Cb are required for connection to link power networks

Figure 5.6 Twisted Pair Termination Network

5.4.3 Grounding Shielded Twisted Pair Cable

When using shielded twisted pair, terminate the twisted pair **and** ground the cable shield, as shown in Figure 5.7.

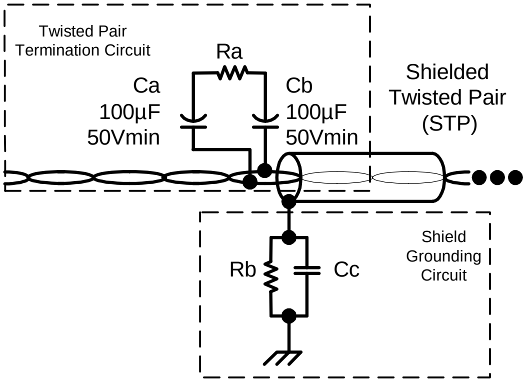


Figure 5.7 Terminating and Grounding Shielded Twisted Pair Cable

The twisted pair is terminated according to the guidelines listed in the previous sections. The cable shield should be grounded using a capacitor to tie the shield to earth ground, and a large-value resistor to bleed off any static charge on the shield.

Tie the shield to earth ground through a capacitor, instead of a direct connection, in order to avoid DC and 50/60 Hz ground paths from being formed through the shield. Typical values for Rb and Cc are as follows:

Cc = 0.1 µF, 10%, Metalized Polyester, ≥ 100V

Rb = 470kΩ, 1/4W, ±5%

The cable shield should be grounded at least once per segment, and preferably at each device.

6

Programming Considerations

6.1 Application Program Development and Export

Applications are initially developed, tested, and debugged using the LonBuilder Developer's Kit or the NodeBuilder Development Tool. See the *LonBuilder User's Guide* or the *NodeBuilder User's Guide* for detailed instructions on developing and testing applications. For updates to the development procedures, also see the ReadMe.txt file installed by the latest service pack for the development tool. Actual unit and system testing of an application targeted for an FT 3120 or FT 3150 Smart Transceiver-based device requires one or more Echelon Model 77040 FTM-10 SMX™ Standard Modular Transceivers combined with an Echelon Model 25000 Neuron Emulator inside of a LonBuilder Development Station, or an Echelon Model 65150-24P LTM-10A/FT-10 Platform. The following two sections describe specifics related to each development environment.

6.2 LonBuilder Developer's Kit

6.2.1 Development Hardware Setup

The ideal development environment connects all LonBuilder Neuron Emulators to the TP/FT-10 twisted pair channel to enable system test and protocol analysis of the functioning control network. The FTM-10 attaches to LonBuilder Neuron Emulators using the Echelon Model 27100 SMX Adapter mounted on the transceiver expansion connectors. Refer to the *LonBuilder Hardware Guide* and *LONWORKS SMX Transceiver Installation Instructions* for detailed instructions on installing the SMX adapter and FTM-10 transceiver.

Warning: A common mistake is to forget to reposition the backplane transceiver jumpers on the LonBuilder Emulator to the *External Transceiver* setting prior to installing the SMX adapter. Also, confirm jumper settings for the FTM-10 transceiver prior to installing on the SMX adapter.

When building an application for FT 3120 or FT 3150 Smart Transceiver-based devices, the LonBuilder communications port settings must use the standard TP/FT-10 definition. If this is not done prior to building the project, the resulting application will be incompatible with the Smart Transceiver release hardware. The first step in defining the communications port settings for a device is to create a new channel definition for TP/FT-10, which is accessed by clicking the **Network** button and then the **Channel** button in the LonBuilder Navigator. Create a channel definition which uses **TP/FT-10** as the **Std Xcvr Type** with **Enforce Std Type** set to **Yes**. Versions of the LonBuilder tool prior to Release 3.01 Service Pack 2 do not include the standard transceiver definition for TP/FT-10. Details of the standard transceiver definition are listed in Table 6.1.

Table 6.1 Standard TP/FT-10 Channel Definition for Bus and Free Topologies

Variable	**TP/FT-10 Standard Transceiver Type**
Comm Mode	Single-ended (see note 4)
Comm Rate	78.13kbps
Min Clock Rate	5MHz
Num Priorities	4
Osc Accuracy	200ppm
Osc Wakeup	0µsec
Avg Packet Size	15 bytes
Collision Detect	No
CD terminate after preamble	No
CD through packet end	No
Bit Sync Threshold	4.0 bits
Rcv Start Delay	9.0 bits (see note 2)
Rcv End Delay	0.0 bits
Indeterm Time	24.0 bits
Min Interpacket Time	0.0 bits
Turnaround	0 µsec
Missed Pream	4.0 bits (see note 3)
Use Raw Data?	No

Note 1) For the Following, "N" must be 1 for the interoperable LONMARK TP/FT-10 channels.

Note 2) For N repeaters in a packet path: Rcv Start Delay = 4.5 * (N + 1) bits.

Note 3) For N repeaters in a packet path: Missed Pream = 2.0 * (N + 1) bits.

Note 4) Use Differential mode when emulating a TP/FT-10 channel on the LonBuilder backplane.

As an alternative to using FTM-10 transceivers, the backplane in the LonBuilder Development Station can approximate the network performance of a Smart Transceiver. This is accomplished by creating a Channel with TP/FT-10 selected as the transceiver type, setting **Enforce Std Type** to **No** and changing the **Comm Mode** type to **Differential**.

Once the TP/FT-10 channel definition has been created, the hardware definition for the target platform must be defined. During development, the LonBuilder Emulator can be used to emulate the memory map of either Smart Transceiver. A minimum version of LonBuilder 3.01 Service Pack 5 is required to build applications for either the FT 3120 or FT 3150 Smart Transceiver. To define a hardware template click the **App Node** button and then the **Properties** button from the LonBuilder Navigator. For FT 3120 Smart Transceiver-based designs, create a hardware template using **3120E4** in Neuron **Chip**. For FT 3150 Smart Transceiver-based designs, create a hardware template using **3150** in **Neuron Chip**. Table 6.2 shows the hardware properties for a typical FT 3150 Smart Transceiver-based custom device.

Table 6.2 Typical LonBuilder Hardware Template Values for an FT 3150 Smart Transceiver Based Device

HW Property Name	**Value**
Neuron Chip	3150
Input Clock Rate	10MHz
ROM Size	0
Flash Size	128 pages
RAM Size	0

The final step in the process is the selection of a target hardware platform. In the development phase, a LonBuilder Emulator is used. To select this target hardware, click the **App Node** button and then the **Target HW** button from the LonBuilder Navigator. Create a target hardware template that uses **Emulator** as the **HW Type**, the TP/FT-10 channel definition created earlier (described above) as the **Channel Name**, and the hardware template created earlier (described above) as the **HW Prop. Name**.

The LonBuilder environment is now ready for application code to be written, built, and loaded on the development platform.

6.2.2 Release Hardware Setup

The only difference between the development hardware setup and the release hardware setup is the selection of the target hardware in the final step. The channel definition and hardware template used in the development hardware setup may be used to build the release version of the application.

The change to the target hardware step is not a requirement, because the programmable application image (.NXE file) from the development build can be used to program the FT 3120 Smart Transceiver internal application space or the external memory of the FT 3150 Smart Transceiver. However, for FT 3120 Smart Transceiver designs and FT 3150 Smart Transceiver designs that use flash memory to store the application, the LonBuilder tool can load the application image over the network.

To select the release hardware as a target for a build, click the **App Node** button and then the **Target HW** button from the LonBuilder Navigator. Create a target hardware template that uses **Custom Node** as the **HW Type**, the TP/FT-10 channel definition created earlier (described above) as the **Channel Name**, and the hardware template created earlier (described above) as the **HW Prop. Name**.

Refer to the *Building Custom Node* section of Chapter 7 of the *LonBuilder User's Guide* for a description of how to migrate the application from LonBuilder Emulators to actual Smart Transceiver-based custom devices.

Warning: In custom designs using flash memory, the programmed device must be explicitly secured by the PROM programmer once the image is programmed. See Software Data Protection (SDP) information provided by the manufacturer of the PROM programmer.

Warning: All FT 3120 Smart Transceiver ICs must be programmed with the generated EEPROM application image file (.nei extension), or its corresponding flash application image file (.nfi extension), before they are soldered onto a PCB assembly.

6.3 NodeBuilder Development Tool

6.3.1 Development Hardware Setup

The NodeBuilder 3 Development Tool must be configured to use the TP/FT-10 channel by using an LNS compatible network interface with a TP/FT-10 connection and installing an FTM-10 SMX transceiver on the LTM-10A Platform. Initial device development is done using the LTM-RAM device template to target the LTM-10A Platform for the execution environment. The LTM-10A platform has a 10MHz input clock. If the hardware is set to run at a lower input clock rate, the design of the application must be tolerant of the reduced execution performance.

Node Builder 3 (Service Pack 1 or later) is required to build applications for the FT 3120 and FT 3150 Smart Transceivers. For detailed instructions on creating this setup using the NodeBuilder tool, refer to the *Quick Start Guide* in the *NodeBuilder User's Guide* and follow the steps below.

The NodeBuilder Device Template wizard runs during the creation of a new device template. This wizard provides an opportunity to select predefined hardware templates. At a later point in the development process, the hardware template can be accessed through the Development and Release folders of the project pane.

During initial device development, use the LTM-10A RAM hardware template. If this template was not selected in the NodeBuilder Device Template Wizard, you can access it by dragging the LTM-10A RAM icon from the Standard Templates folder located in the Hardware Templates folder to the device template Development folder for the device.

6.3.2 Release Hardware Setup

Once the application has been developed, tested, and debugged on the LTM-10A platform, a hardware template must be generated to match the final target hardware. Refer the *NodeBuilder User's Guide* and follow the steps listed in this section, to generate the device template.

Predefined hardware templates for the FT3120 and FT3150 Smart Transceivers are included with the NodeBuilder software. These templates may be used to describe the release hardware or as a reference point to create a customized template.

When using the FT3120 Smart Transceiver at 40MHz, use the FT 3120-E4 40MHz device template. During the initial process of creating a device in the NodeBuilder software, select this template in the NodeBuilder Device Template wizard. To access the template later, drag the FT 3120-E4 40MHz icon from Standard Templates folder into the Release folder of the device. Similarly, applications developed for the FT3150 Smart Transceiver that utilize a 64K external memory part should use the predefined FT 3150 64K Flash 10MHz standard template.

If the final target device does not match the predefined templates, create a hardware template. First, drag the appropriate FT standard template into the User Templates folder in the Hardware Templates folder of the Project pane. For a FT3120 Smart Transceiver, use the FT 3120-E4 40MHz device template. For a FT3150 Smart Transceiver, use the FT 3150 64K Flash 10MHz device template.

Once the standard template icon has been placed in the User Templates folder, the NodeBuilder Hardware Template Properties window opens with the values of the selected standard template. Change the template's name and modify the template properties as follows:

Specify the target hardware clock speed in **Clock Speed** on the Hardware tab. Specify the address map on the Off-chip Memory tab for the device. For example, a 32K part will have **Non-volatile End** set to 7FFF and all other fields set to 0. Also, select the memory part **Type**. For memory parts, you must specify the sector size. For EEPROM memory, you must set the write time. A list of Echelon tested external memory components can be found on the Echelon website at www.echelon.com.

Once the hardware template has been created, drag the new hardware template to the device. Drag the newly created icon in the User Templates folder to the Release folder of the device template to complete the procedure.

Note: You can update the user-defined hardware template later by double-clicking the template icon. Your new changes will affect any projects opened and compiled using this template. A Build All may be required if you change a hardware template without making any other changes.

Warning: Flash memory must be explicitly secured by the PROM programmer once the image is programmed. See SDP information provided by the manufacturer of the PROM programmer.

Warning: All FT 3120 Smart Transceivers must be programmed with the generated EEPROM application image file (.nei extension), or the corresponding flash application image file (.nfi extension), before they are soldered onto a PCB assembly.

A

FT 3120 / FT 3150 Smart Transceiver Design Checklist

This appendix includes a checklist to ensure that products using the FT 3120 and FT 3150 Smart Transceivers meet the specifications presented in this user's guide.

A.1 FT 3120 / FT 3150 Smart Transceiver-based Device Checklist

Table A.1 FT 3120 / FT 3150 Smart Transceiver Connections

Item	Check When Completed	Description
1		FT-X1 transformer pins connected as shown in the *FT 3120 / FT 3150 Smart Transceiver Datasheet* and in Figure 4.1.
2		Environmental and electrical specifications as shown in the *FT 3120 / FT 3150 Smart Transceiver Datasheet*.
3		Pin T1 from the Smart Transceiver is connected to pin 3 on the FT-X1 transformer.
4		Pin T2 from the Smart Transceiver is connected to pin 4 on the FT-X1 transformer.
5		The recommended number and placement of 0.1µF bypass capacitors are near the Smart Transceiver.
6		The Smart Transceiver input clock frequency is 5MHz, 10MHz, 20MHz, or 40MHz and accurate to ± 200 ppm. The input clock frequency of 40MHz is valid for the FT 3120 Smart Transceiver only.
7		The Smart Transceiver and FT-X1 Communications Transformer are placed adjacent to one another on the same printed circuit board.
8		If required, a low voltage interrupt (LVI) circuit with open collector output is used to supply a reset signal to the Smart Transceiver.
9		Clamping diodes and 470V MOV are used in the ESD protection circuit shown in Figure 4.1.

Table A.2 FT 3120 / FT 3150 Smart Transceiver PCB Layout

Item	Check When Completed	Description
10		Star ground configuration used.
11		Keepout areas observed for PCB.
12		Ground planes or wide traces used to lower inductance of the ground system.
13		The leakage capacitance from high frequency circuit traces is controlled via guard traces.
14		The product's package is designed to minimize the possibility of ESD hits arcing into the device's circuit board. If the product's package is plastic, then the PCB is supported in the package so that unprotected circuitry on the PCB is not adjacent to any seams in the package. The PCB is not touching the plastic enclosure near a seam.
15		Explicit clamping of user-accessible circuitry is used to shunt ESD currents from that circuitry to the center of the star ground on the PCB.
16		The network connector, diodes and MOV shown in Figure 4.1 are all located close to the center of the star ground.

Table A.3 FT 3120 / FT 3150 Smart Transceiver Programming

Item	Check When Completed	Description
17		TP/FT-10 is used as the channel definition in the development tool.
18		If using LonBuilder 3.0.1 or NodeBuilder 1.5 and running at a clock speed greater than 10MHz, use the latest available service pack for the development tool. Consult the readme.txt file for further programming considerations.

Table A.4 FT 3120 / FT 3150 Smart Transceiver Power Supply - General

Item	Check When Completed	Description
19		Supply provides 4.75VDC minimum, 5.25VDC maximum.

B

Qualified TP/FT-10 Cable Specifications and Sources

This appendix documents generic cable specifications that have been qualified by Echelon to work with TP/FT-10 channels. Specific vendors and their cables are cited to highlight the variety of cable types available that meet these generic specifications.

B.1 Five Qualified Cables

Echelon has qualified five cables that are available from a large number of different vendors. Table B.1 describes these cables.

Table B.1 Qualified Cables

Cable Type	**AWG**	**Diameter**	**Comment**
TIA568A Category 5 cable	24AWG	0.5mm	Widely available, and can be found as part of structured cabling systems such as Lucent's Systimax®. See also section B.1.2, *NEMA Level IV Cable Specifications*.
NEMA Level IV cable	16AWG	1.3mm	Available with a broad range of options, including stranded or solid, 1 or 2 pairs per cable, shielded or unshielded, and plenum or PVC. See section B.1.3, *16AWG/1.3mm "Generic" Cable Specifications*.
Belden 8471 cable	16AWG	1.3mm	See section B.1.3, *16AWG/1.3mm "Generic" Cable Specifications*.
Belden 85102 cable	22AWG	0.65mm	See section B.1.2, *NEMA Level IV Cable Specifications*.
JY (st) Y 2x2x0.8 cable	20.4AWG	0.8mm	Available in Europe only. Please see he *Junction Box and Wiring Guidelines for Twisted Pair LONWORKS Networks* engineering bulletin (005-0023-01) for more information.

A list of cable vendors for each cable type can be found in the *Junction Box and Wiring Guidelines for Twisted Pair LONWORKS Networks* engineering bulletin (005-0023-01). This document is available from Echelon's web site at www.echelon.com.

B.1.1 Category 5 Cable Specifications

The specification for the Telecommunications Industry Association's Commercial Building Telecommunications Cabling Standard (ANSI/TIA/EIA-568-A-95) is available from Global Engineering Documents (http://global.his.com).

B.1.2 NEMA Level IV Cable Specifications

The Level IV cable specification used by Echelon and as originally defined by the National Electrical Manufacturers Association (NEMA) differs from the Category IV specification proposed by the Electronic Industries Association/Telecommunication Industry Association (EIA/TIA). The Level IV cable specifications used by Echelon are presented below, and are followed by a list of Level IV cable suppliers.

In the following tables, Specifications apply to shielded or unshielded 22AWG (0.65mm) cable, 24AWG (0.5mm) cable shown in brackets [ ] if different.

Table B.2 General Specifications

Specification	**Value**
DC Resistance (Ohms/1000 feet at 20°C) maximum for a single copper conductor regardless of whether it is solid or stranded and is or is not metal-coated.	18.0 [28.6]
DC Resistance Unbalance (percent) maximum	5
Mutual Capacitance of a Pair (pF/foot) maximum	17
Pair-to-Ground Capacitance Unbalance (pF/1000 feet) maximum	1000

Table B.3 Impedance Characteristics

Frequency	Impedance (Ohms)
772kHz	102±15% (87-117)
1.0MHz	100±15% (85-115)
4.0MHz	100±15% (85-115)
8.0MHz	100±15% (85-115)
10.0MHz	100±15% (85-115)
16.0MHz	100±15% (85-115)
20.0MHz	100±15% (85-115)

Table B.4 Attenuation (dB/1000 feet at 20°C) Maximum

Frequency	Attenuation
772kHz	4.5 [5.7]
1.0MHz	5.5 [6.5]
4.0MHz	11.0 [13.0]
8.0MHz	15.0 [19.0]
10.0MHz	17.0 [22.0]
16.0MHz	22.0 [27.0]
20.0MHz	24.0 [31.0]

Table B.5 Worst-Pair Near-End Crosstalk (dB) Minimum

Frequency	Crosstalk db
772kHz	58
1.0MHz	56
4.0MHz	47
8.0MHz	42
10.0MHz	41
16.0MHz	38
20.0MHz	36

Values are shown for information only. The minimum near-end cross talk (NEXT) coupling loss for any pair combination at room temperature is to be greater than the value determined using the formula for all frequencies in the range of 0.772MHz-20MHz for a length of 1000 feet:

$$\mathrm{NEXT}\ (F_{MHz})>\mathrm{NEXT}(0.772)-15*\log(F_{MHz}/0.772)$$

B.1.3 16AWG/1.3mm “Generic” Cable Specifications

The specifications for the 16AWG/1.3mm Generic cable qualified by Echelon for use with TP/FT-10 networks is presented below. The generic single twisted pair is stranded (19 x 29) with tinned copper.

Table B.6 General Specifications

Specification	Minimum	Typical	Maximum	Units	Condition
DC Resistance, each conductor	14.0	14.7	15.5	Ω/km	20°C per ASTM D 4566
DC Resistance Unbalance			5%		20°C per ASTM D 4566
Mutual Capacitance			55.9	nF/km	Per ASTM D 4566
Characteristic Impedance	92	100	108	_	64kHz to 1MHz, per ASTM D 4566

Table B.7 Attenuation and Propagation Delay Characteristics

Characteristic	Minimum	Typical	Maximum	Units	Condition
Attenuation				dB/km	20°C per ASTM D 4566
20kHz			1.3		
64kHz			1.9		
78kHz			2.2		
156kHz			3.0		
256kHz			4.8		
512kHz			8.1		
772kHz			11.3		
1000kHz			13.7		
Propagation Delay			5.6	ns/m	78kHz

C

Design and Handling Guidelines

C.1 Application Considerations

C.1.1 Termination of Unused Pins

Because the FT 3120 and FT 3150 Smart Transceivers are CMOS devices, unused input pins *including undeclared/unconnected I/O pins configured as inputs or three-state* must be terminated to assure proper operation and reliability. Figure C.1 shows a CMOS inverter representative of circuitry found on CMOS input pins. When the input is logic zero, the P-channel transistor is on (conducts), and the N-channel transistor is off. When the input is a logic one, the P-channel transistor is off, and the N-channel transistor is on. These transistors are linear devices with relatively broad switch points. As the input transitions through the mid-supply region, there is a duration of time when both transistors are conducting. With fast rise time digital signals at the input, this duration is very short. Once the inverter is out of the linear region there is very little current flow. This effect is the reason that the overall current drain of a CMOS device is directly proportional to the switching speed. Almost all the current consumption is by transistors passing through the linear region and charging and discharging of internal capacitances. If a pin is configured as an input or three-state, then the input can oscillate due to supply noise or float to the mid-supply region, resulting in higher current consumption. Current design techniques have made latch-up due to floating input unlikely, but it is good design practice to terminate unused I/O pins that are not configured (high impedance) or are configured as inputs. On the Smart Transceiver, the only pin other than the I/O pins that could be configured as an unterminated input, is the $\overline{\text{SERVICE}}$ pin if the optional pull-up is disabled. If the optional pull-up devices are disabled on IO4 – IO7, then termination is necessary for those pins. Pull-ups are enabled in Neuron C with the `#pragma enable_io_pullups` compiler directive.

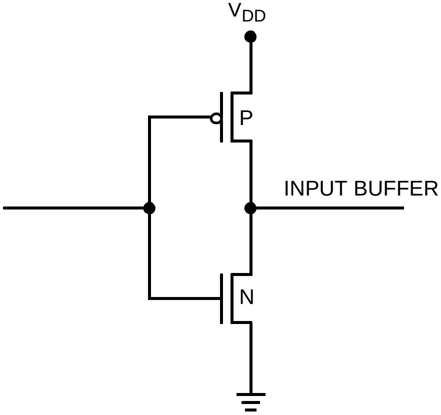


Figure C.1 CMOS Inverter

The best method to terminate unused I/O pins is with an individual pull-up or pull-down resistor for each unused pin. Unused input pins can be connected to each other and then to a common termination point. This cost/space effective method has the disadvantage of not allowing individual pin configuration later and the possibility of contention in the event pins are declared as outputs. Individual unused I/O pins may be connected directly to V_{SS} or V_{DD}, but this is not recommended in case of software error and the possibility of output declaration to an opposing state. Unused pins may be declared as outputs, but this consumes application code space which is particularly valuable in FT 3120 Smart Transceiver applications. A pin capable of being configured as an output should never be connected to another such pin or directly to V_{SS} or V_{DD}.

C.1.2 Avoidance of Damaging Conditions

All integrated circuit devices can be damaged or destroyed by exceeding specified voltage and environmental limits. These limits are conservative to ensure reliable operation within the conditions specified.

Most potentially destructive AC waveforms fall into one of two categories. One type is high-voltage (10kV – 25kV), low-energy spikes usually under 100ns in duration due to ESD discharge. ESD modeling has shown that the human body can generate and discharge electrostatic voltages of up to 12kV. The second type is lower-voltage, higher-energy transients that can last for several hundred microseconds or more and can be caused by capacitive coupling of lightning or inductive load sources. Different protection devices must be implemented, depending on what is anticipated in the operating environment. Failure modes can be quantified and protective precautions taken to avoid product malfunction. This may be PC board layout-related or may involve the use of external protection devices. All pins on the Smart Transceiver have internal diode protection that will protect ESD type transients up to 2.2kV. External protection is required in products subject to human contact or where interfaces to other equipment may be encountered.

Many factors, including ambient temperature and semiconductor lot-to-lot processing variations, will influence the effect of illegal conditions on the Smart Transceiver. The V_{SS} ground pins are internally connected to the substrate of the silicon die and are the reference point for all voltages. The Smart Transceiver functions for a V_{DD} connected to the positive supply pin(s). In limited temperature range environments, the device may operate over a wider V_{DD} with timing, drive, FT transceiver and other specifications not met. There may also be some adverse effects on gate oxides from long-term exposures to V_{DD} greater than 5.5V.

Zap and *latch-up* refer to two damage mechanisms resident in CMOS ICs. *Zap* refers to damage caused by very-high-voltage, static-electricity exposure. This damage usually appears as breakdown of the relatively thin oxide layers that causes leakage or shorts. Often secondary damage occurs after an initial zap failure causes a short.

Latch-up refers to a usually catastrophic condition that is caused by turning on a parasitic, bipolar, silicon-controlled rectifier (SCR). A latch-up is formed by N and P regions in the layout of the integrated circuit, which act as the collector, base, and emitter of parasitic transistors. Bulk resistance of silicon in the wells and substrate acts as resistors in the SCR circuit. Application of voltages to pins above V_{DD} + 0.3V or below V_{SS} – 0.3V in conjunction with enough current to develop voltage drops across the parasitic resistors can cause the SCR to turn on. Once on, the SCR can be turned off only by removal of all power and applied voltages. The low on-impedance of the SCR circuit can overheat and destroy the IC.

Figure C.2 shows the MOS circuitry for a digital input-only pin. The gates of the input buffer are very high impedance for all voltages that would ever be applied to the pin. Protection is implemented with a P-channel transistor acting as a diode to V_{DD} and an N-channel transistor acting as a diode to V_{SS}. Allowing a pin to float or be driven to a mid-supply level can result in both the N- and P-channel devices in the input buffer simultaneously being partially on, which causes excess current and noise on the V_{DD}/V_{SS} power supply. If a digital input is driven above V_{DD}, the pseudo-diode will conduct, protecting the input. As the current is increased to high levels (100mA), damage can result. Figure C.3 shows the CMOS circuitry for a digital input/output-only pin.

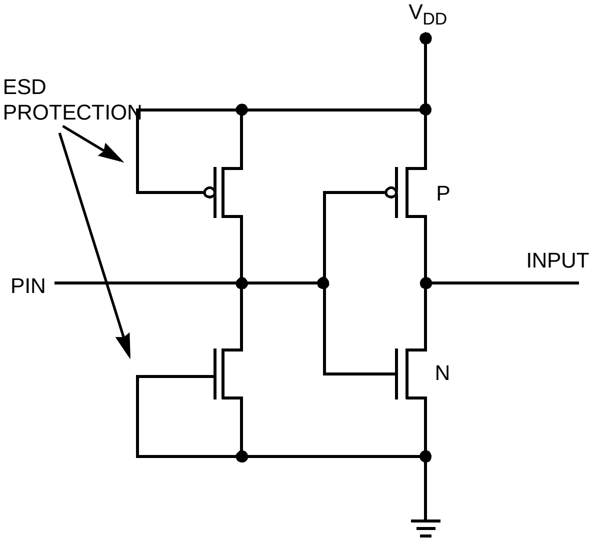


Figure C.2 Digital Input

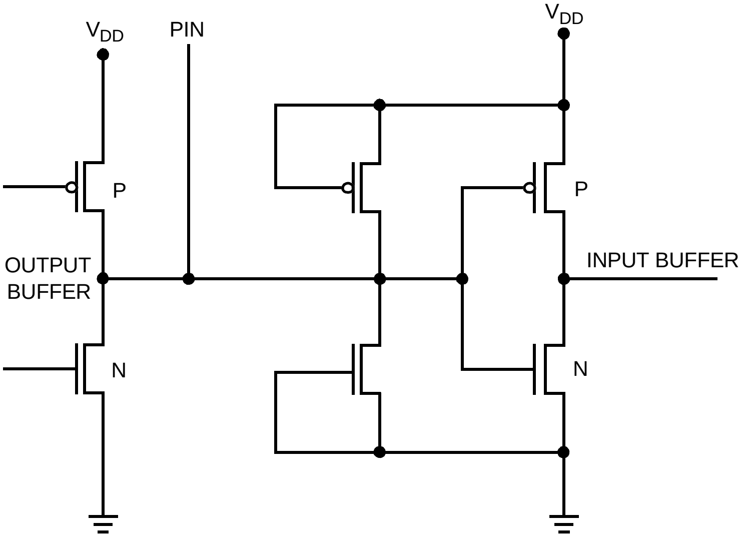


Figure C.3 Digital I/O

C.1.3 Power Supply, Ground, and Noise Considerations

The most important considerations for PCB layout deal with noise. This includes noise on the power supply, noise generated by the digital circuitry on the device, and coupling digital signals into the analog signals. The best PCB layout methods to prevent noise induced problems are the following:

- Keep digital signals as far away from analog signals as possible.
- Use short, low-inductance traces for the analog circuitry to reduce inductive, capacitive, and radio frequency noise sensitivities.
- Use short, low-inductance traces for the digital circuitry to reduce inductive, capacitive, and radio frequency radiated noise.
- Bypass capacitors should be connected between the V_{DD} and V_{SS} pairs with minimal trace length. These capacitors help supply the instantaneous currents of the digital circuitry in addition to decoupling the noise that may be generated by other sections of the device or other circuitry on the power supply.
- Use short, wide, low-inductance traces to connect all of the V_{SS} ground pins together. Depending on the application, a double-sided PCB with a V_{SS} ground plane under the device connecting all of the digital and analog V_{SS} pins together would be a good grounding method. A multilayer PCB with a ground plane connecting all of the digital and analog V_{SS} pins together would be the optimal ground configuration. These methods will result in the lowest resistance and the lowest inductance in the ground circuit. This is important to reduce voltage spikes in the ground circuit resulting from the high-speed digital current spikes. Suppressing these voltage spikes on the integrated circuit is the reason for multiple V_{SS} ground leads.
- Use short, wide, low-inductance traces to connect all of the V_{DD} power supply pins together. Depending on the application, a double-sided PCB with V_{DD} bypass capacitors to the V_{SS} ground plane under the device may complete the low-impedance coupling for the power supply. For a multilayer PCB with a power plane, connecting all of the digital and analog V_{DD} pins to the power plane would be the optimal power distribution method. The integrated circuit layout and packaging considerations for the 5V V_{DD} power circuit are essentially the same as for the ground circuit.

C.1.4 Decoupling Capacitors

The switching waveforms shown in Figures C.4 and C.5 show the current spikes introduced to the power supply and ground lines. This effect is shown for a load capacitance of less than 5pF or for 50pF. For ideal power supply lines with no series impedance, the spikes would pose no problem. However, actual power supply and ground lines do possess series impedance, giving rise to noise problems. For this reason, care should be taken in board layouts, ensuring low-impedance paths to and from logic devices.

To absorb switching spikes, the following CMOS devices should be bypassed with good quality 0.022µF to 0.33µF decoupling capacitors:

- Bypass every device driving a bus with all outputs switching simultaneously.
- Bypass all synchronous counters.
- Bypass devices used as oscillator elements.
- Bypass Schmitt-trigger devices with slow input rise and fall times. The slower the rise and fall time, the larger the bypass capacitor. Lab experimentation is suggested.

Bypass capacitors should be distributed over the circuit board. In addition, boards could be decoupled with a 1 µF capacitor.

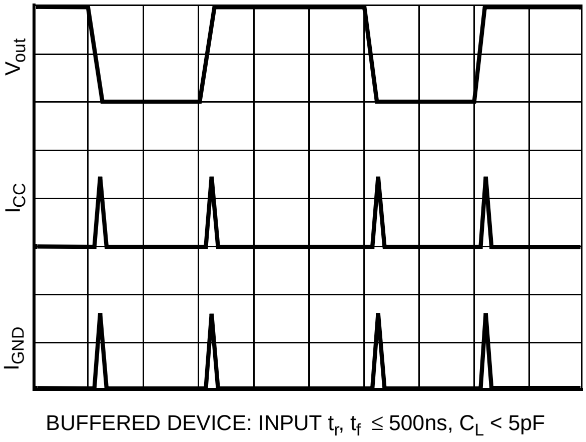

V_{out}
I_{CC}
I_{GND}
BUFFERED DEVICE: INPUT t_r, t_f $\le$ 500ns, C_L < 5pF

Figure C.4 Switching Currents for C_L < 5pF

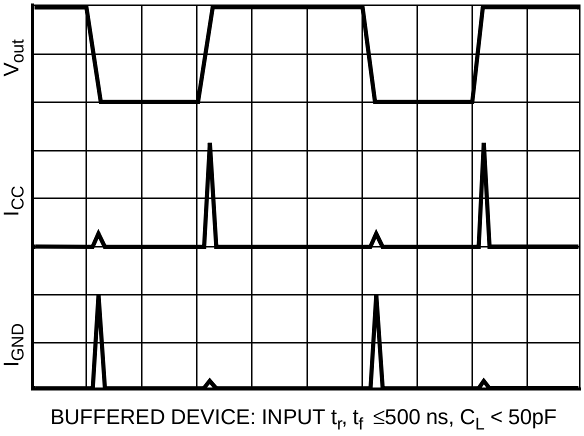


Figure C.5 Switching Currents for C_L = 50pF

C.2 Board Soldering Considerations

Dry pack is a process which slowly bakes moisture from the surface mount technology (SMT) package and then seals it into a dry pack bag to shield the unit from moisture in the atmosphere. The exterior of the bag will be marked with a label that indicates the devices are moisture sensitive and is marked with the date the bag was sealed (there is a one-year shelf life for these devices). There is a limited amount of time to use surface-mount devices once they are removed from the dry pack. Before surface mounting, packages should not be out of the dry pack longer than 168 hours at < 60% relative humidity and < 30°C. If the units have not been shipped dry pack or have been unpacked for too long, then units must be baked at 125°C for 6 hours prior to board soldering. The old recommendation was 24 hours at 125°C. If this is not done, some percentage of the units will exhibit destructive failures or latent failures after the soldering process.

C.3 Handling Precautions and Electrostatic Discharge

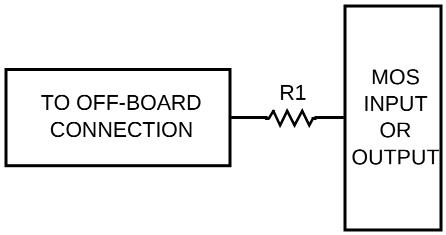
All CMOS devices have an insulated gate that is subject to voltage breakdown. The gate oxide for the Smart Transceiver breaks down at a gate-source potential of about 10V. The high-impedance gates on the devices are protected by on-chip networks. However, these on-chip networks do not make the IC immune to ESD. Laboratory tests show that devices may fail after one very high voltage discharge. They may also fail due to the cumulative effect of several discharges of lower potential.

Static-damaged devices behave in various ways, depending on the severity of the damage. The most severely damaged are the easiest to detect because the input or output has been completely destroyed and is shorted to V_{DD}, shorted to V_{SS}, or is open-circuited. As a result of this, the device will no longer function. Less severe cases are more difficult to detect because they appear as intermittent failures or degraded performance. Static damage can often increase leakage currents.

CMOS devices are not immune to large static voltage discharges that can be generated while handling. For example, static voltages generated by a person walking across a waxed floor have been measured in the 4 kV – 15 kV range (depending on humidity, surface conditions, etc.). Therefore, the following precautions should be observed.

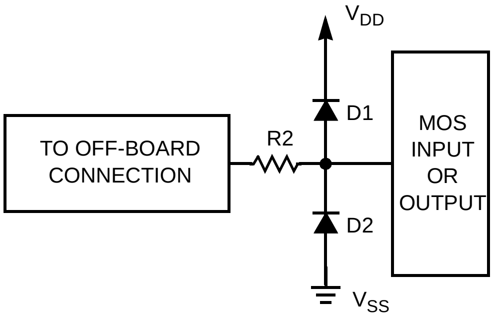
1. Do not exceed the maximum ratings specified by the data sheet.
2. All unused device inputs should be connected to V_{DD} or V_{SS}.
3. All low-impedance equipment (pulse generators, etc.) should be connected to CMOS inputs only after the device is powered up. Similarly, this type of equipment should be disconnected before power is turned off.
4. A circuit board containing CMOS devices is merely an extension of the device and the same handling precautions apply. Contacting connectors wired directly to devices can cause damage. Plastic wrapping should be avoided. When external connections to a PC board address pins of CMOS integrated circuits, a resistor should be used in series with the inputs or outputs. The limiting factor for the series resistor is the added delay caused by the time constant formed by the series resistor and input capacitance. This resistor will help limit accidental damage if the PC board is removed and is brought into contact with static-generating materials. For convenience, equations for added propagation delay and rise-time effects due to series-resistance size are given in Figure C.6.

5. All CMOS devices should be stored or transported in materials that are antistatic. Devices must not be inserted into conventional plastic “snow,” Styrofoam®, or plastic trays. Devices should be left in their original container until ready for use.
6. All CMOS devices should be placed on a grounded bench surface and operators should ground themselves prior to handling devices, since a worker can be statically charged with respect to the bench surface. Wrist straps in contact with skin are strongly recommended. See Figure C.7.
7. Nylon or other static-generating materials should not come in contact with CMOS circuits.
8. If automatic handling is being used, high levels of static electricity may be generated by the movement of devices, belts, or boards. Reduce static build-up by using ionized air blowers or room humidifiers. All parts of machines which come into contact with the top, bottom, and sides of IC packages must be grounded metal or other conductive material.
9. Cold chambers using CO_2 for cooling should be equipped with baffles, and devices must be contained on or in conductive material, or soldered onto a PCB.
10. When lead-straightening or hand-soldering is necessary, provide ground straps for the apparatus used and be sure that soldering ties are grounded.
11. The following steps should be observed during wave-solder operations.
    - a. The solder pot and conductive conveyor system of the wave-soldering machine must be grounded to an earth ground.
    - b. The loading and unloading work benches should have conductive tops which are grounded to an earth ground.
    - c. Operators must comply with precautions previously explained.
    - d. Completed assemblies should be placed in antistatic containers prior to being moved to subsequent stations.



Advantage: Requires minimal board area.

Disadvantage: R1 > R2 for the same level of protection, therefore rise and fall times, propagation delays, and output drives are severely affected.



Advantage: R2 < R1 for the same level of protection. Impact on ac and dc characteristics is minimized.

Disadvantage: More board area, higher initial cost.

NOTE: These networks are useful for protecting the following:

a. digital inputs and outputs
b. analog inputs and outputs
c. three-state outputs
d. bidirectional (I/O) ports

Equation 1 – Propagation Delay vs. Series Resistance

$$R \approx \frac{t}{C \bullet k}$$

where:

R = the maximum allowable series resistance in ohms
t = the maximum tolerable propagation delay in seconds
C = the board capacitance plus the driven device's input capacitance in farads
k = 0.33 for the TTL input levels (switch point = 1.3 V)

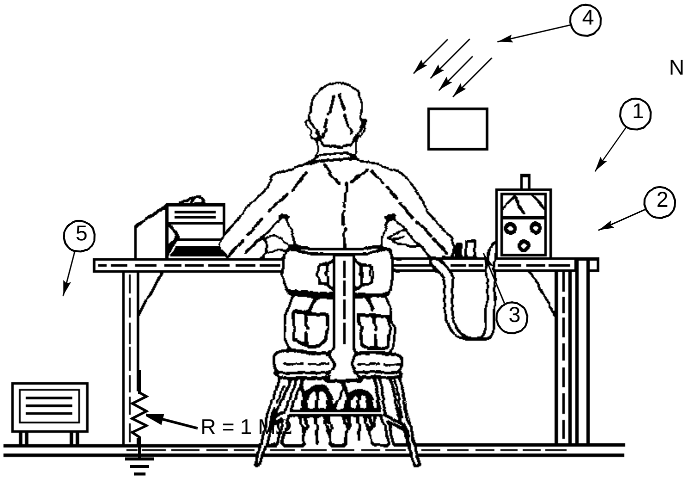
Equation 2 – Rise Time vs. Series Resistance

$$R \approx \frac{t}{C \bullet k}$$

where:

R = the maximum allowable series resistance in ohms
t = the maximum rise time per data sheet in seconds
C = the board capacitance plus the driven device's input capacitance in farads
k = 2.3 for other devices

Figure C.6 Networks for Minimizing ESD and Reducing CMOS Latch-Up Susceptibility



NOTES:
1. 1/16 inch conductive sheet stock covering bench-top work area.
2. Ground strap.
3. Wrist strap in contact with skin.
4. Static neutralizer. (Ionized air blower directed at work.) Primarily for use in areas where direct grounding is impractical.
5. Room humidifier. Primarily for use in areas where the relative humidity is less than 45%. Caution: building heating and cooling systems usually dry the air causing the relative humidity inside a building to be less than outside humidity.

Figure C.7 Typical Manufacturing Work Station

1. The following steps should be observed during board cleaning operation.
   - **a.** Vapor degreasers and baskets must be grounded to an earth ground. Operators must likewise be grounded.
   - **b.** Brush or spray cleaning should not be used.
   - **c.** Assemblies should be placed into the vapor degreaser immediately upon removal from the antistatic container.
   - **d.** Cleaned assemblies should be placed in antistatic containers immediately after removal from the cleaning basket.
   - **e.** High-velocity air movement or application of solvents and coatings should be employed only when module circuits are grounded and a static eliminator is directed at the module.The use of static-detection meters for line surveillance is highly recommended.
   - **f.** The use of static-detection meters for line surveillance is highly recommended.
   - **g.** Equipment specifications should alert users to the presence of CMOS devices and require familiarization with this specification prior to performing any kind of maintenance or replacement of devices or modules.

h. Do not insert or remove CMOS devices from test sockets with power applied. Check all power supplies to be used for testing devices to be certain there are no voltage transients present.

i. Double-check the equipment setup for proper polarity of voltage before conducting parametric or functional testing.

j. Do not recycle shipping rails. Continuous use causes deterioration of their anti-static coating.

k. Wrist straps and equipment logs should be maintained and audited on a regular basis. Wrist straps malfunction and may go unnoticed. Also, equipment gets moved from time to time and grounds may not be reconnected properly.

C.3.1 Electrostatic Discharge

There are many ways to deal with ESD, including:

- Divert or limit energy from points of contact to circuitry.
- Start with a series of electromagnetic interference (EMI) ferrites or resistors for high frequency filtering.
- Use diodes, transient voltage suppressors (ex: MOSorbs, transorbs, ...) for high-speed clamping.
- Use capacitors to protect critical inputs.
- Use good power distribution.
- Use a separate, low-impedance ESD ground path to divert energy from electronics (ex: “star” ground strategy).

NOTE:The impedance of a wire at 300MHz is approximately 20Ω/cm. Use a conductor with less than or equal to 3:1 length:width ratio.

RECOMMENDED READING

Total Control of the Static in Your Business

Available by writing to:

Static Control Systems Div.
Box ELB-3, 225-4S
3M Center
St. Paul, MN 55144

Or calling:

1-800-328-1368
1-612-733-9420 (in Minnesota)

C.4 Power Distribution and Decoupling Capacitors

Inductance in the power distribution creates noise during switching transients.

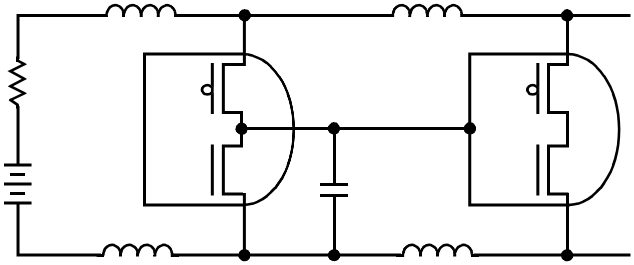


Figure C.8 Inductance Creates Noise

For example, a 200nH inductor with 25mA and 5ns surge characteristic can generate 1 V in noise:

```
Vnoise = L dI/dt
L = 200nH, dI = 25mA, dt = 5ns
Vnoise = 200 nH * 25mA / 5ns = 1V
```

C.5 Recommended Bypass Capacitor Placement

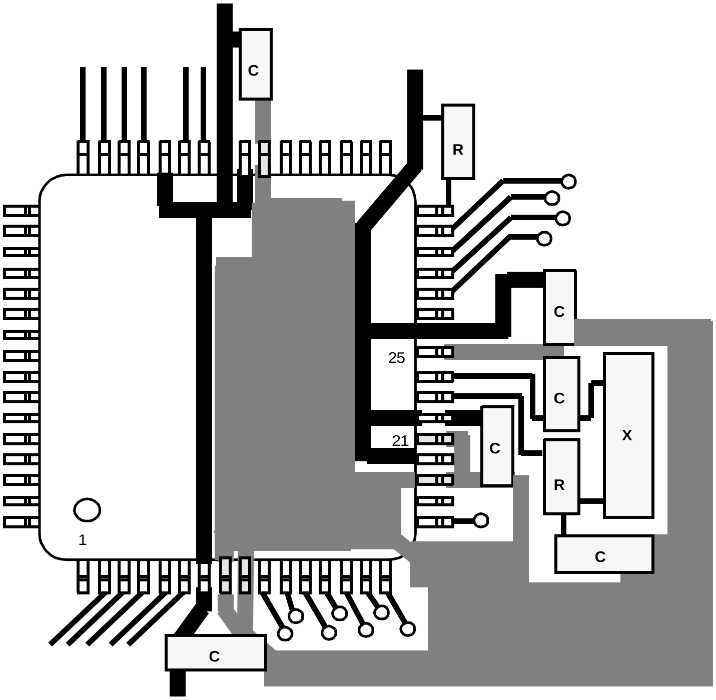
Proper decoupling is required to ensure proper operation of an FT 3120 or FT 3150 Smart Transceiver. When connecting V_{DD} decoupling capacitors to Smart Transceivers, make the leads as short as possible. All V_{DD} pins must be tied to +5V, and all V_{SS} pins to ground. Keep the crystal circuit close to the Smart Transceiver and isolated from communication lines.

Bypass capacitors should be 0.1µF or 0.33µF ceramic or dipped-mica capacitors and should be placed as close to V_{DD} pins as possible. V_{DD} and GND loops should be avoided. Minimum recommended configurations are:

- FT 3150 Smart Transceiver: 0.1µF bypass capacitor between pins: 7:8, 21:22, 25:26, 39:40.
- FT 3120 Smart Transceiver: 0.1µF bypass capacitor between pins: 10:11, 12:13, 16:18, 31:32.

The following figures, Figure C.9, Figure C.10 and Figure C.11, show suggested bypass capacitor placement and crystal circuit trace outlines.

FT 3150 SMART TRANSCEIVER 64-LEAD TQFP



KEY LAYOUT RULES

1. If possible, use 4-layer (or more) boards. This would greatly simplify the layout and reduce grounding and noise-related problems.
2. For 2-layer boards, the four bypass capacitors must be close to the Smart Transceiver IC. V_{DD} and ground must be large traces to reduce inductance and noise.
3. The crystal must be isolated from any digital signal. If clock 2 is being used for other circuit signals, keep the trace short or buffer it. The added board capacitance and input capacitance of other devices being driven will skew the crystal frequency.
4. On 2-layer boards, avoid running high-frequency digital signal traces under crystal circuit or input pins of the communications signals, on opposite sides of the board.
5. Ensure that power supply and ground traces are large enough to handle the peak surge switching currents. Otherwise there will be power supply dips on the V_{DD} pins, which may cause errors in the checksum calculation, resulting in the Smart Transceiver going applicationless.

NOTE:

C = Capacitor, surface mount.
R = Resistor.
X = Crystal.

Figure C.9 Minimum Recommended Capacitor Placement (Sheet 1 of 3)

FT 3120 SMART TRANSCEIVER
32-LEAD SOG

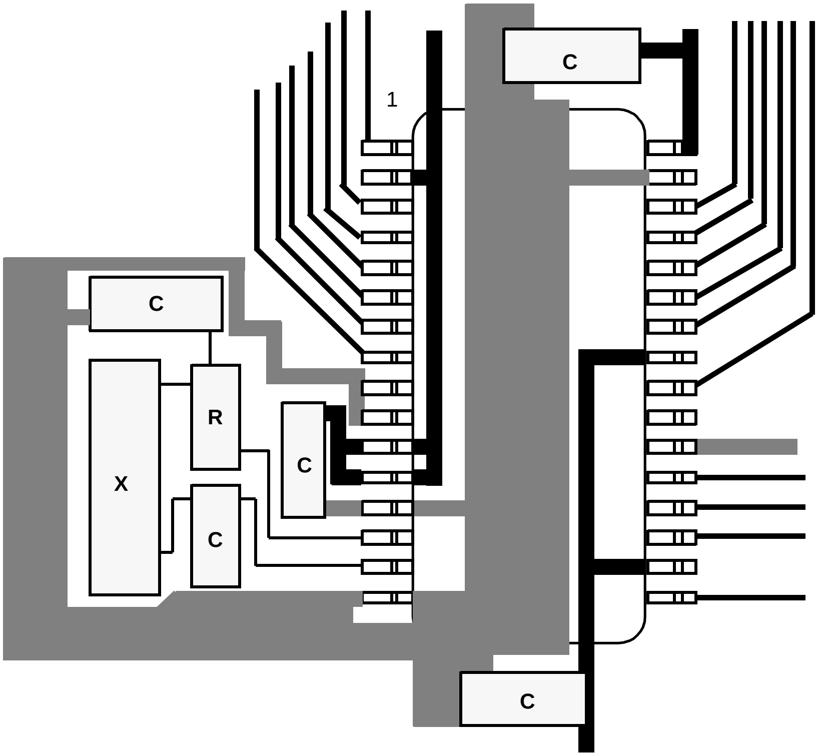


Figure C.10 Minimum Recommended Capacitor Placement (Sheet 2 of 3)

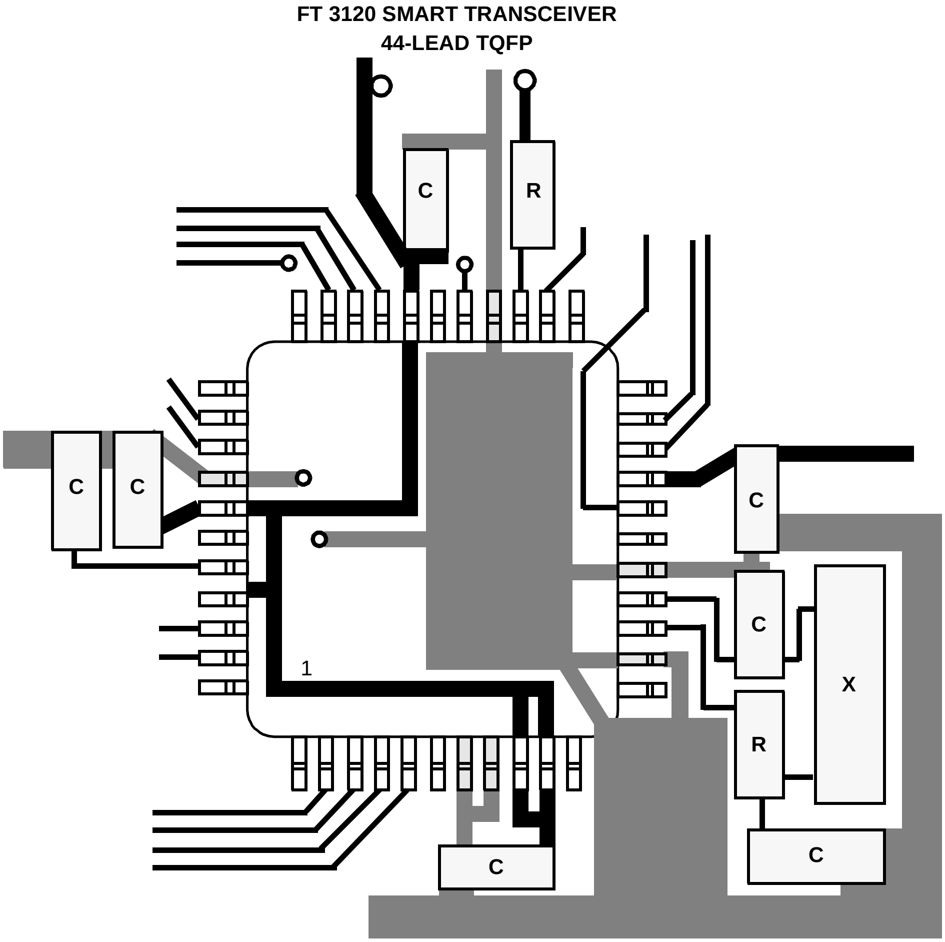


Figure C.11 Minimum Recommended Capacitor Placement (Sheet 3 of 3)

Index

ECHELON®

www.echelon.com