**Welcome to E-XFL.COM**

## Embedded - Microcontrollers - Application Specific: Tailored Solutions for Precision and Performance

**Embedded - Microcontrollers - Application Specific** represents a category of microcontrollers designed with unique features and capabilities tailored to specific application needs. Unlike general-purpose microcontrollers, application-specific microcontrollers are optimized for particular tasks, offering enhanced performance, efficiency, and functionality to meet the demands of specialized applications.

## What Are Embedded - Microcontrollers - Application Specific?

Application-specific microcontrollers are engineered to

| Details | |
|---|---|
| Product Status | Obsolete |
| Applications | Network Processor |
| Core Processor | ARM7® |
| Program Memory Type | External Program Memory |
| Controller Series | NET+40 |
| RAM Size | External |
| Interface | EBI/EMI, Ethernet, DMA, HDLC, IEEE1284/ENI, SPI, UART |
| Number of I/O | 24 |
| Voltage - Supply | 3V ~ 3.6V |
| Operating Temperature | -40°C ~ 85°C |
| Mounting Type | Surface Mount |
| Package / Case | 208-BFQFP |
| Supplier Device Package | 208-PQFP (28x28) |
| Purchase URL | https://www.e-xfl.com/product-detail/digi-international/net-40-qipro-4 |

# Table of Contents

| | |
|---|---|
| **FAULT1*** | **1284 Channel 1 1284 FAULT** |
| **FAULT2*** | **1284 Channel 1 1284 FAULT** |
| **FAULT3*** | **1284 Channel 1 1284 FAULT** |
| **FAULT4*** | **1284 Channel 1 1284 FAULT** |

The channel FAULT inputs are an input to the NET+ARM chip from the external 1284 port. The meaning of the FAULT input changes as the 1284 operational mode changes.

## 1.5.5 ENI Interface Configured for ENI Host Mode

**PCS*  ENI Chip Select**

The PCS* signal provides the chip enable for the ENI interface from the external processor system. When PCS* is driven low, the ENI interface uses the PA address bus to determine which resource is being addressed. After the ENI interface cycle is complete, the ENI interface asserts the PACK* signal to complete the cycle. After PACK* is asserted active, the external processor must de-assert PCS* to complete the ENI interface cycle.

**PRW*  ENI Read/Write**

The PRW* is driven by the external processor system during an ENI access cycle to indicate the data transfer direction. During normal PCS* cycles, PRW* high indicates a READ from the ENI interface; PRW* low indicates a WRITE to the ENI interface. During DMA cycles, PRW* high indicates a WRITE to the ENI interface; PRW* low indicates a READ from the ENI interface.

**PA[16:0]  ENI Address Bus**

The ENI interface requires 17 address signals to identify the resource being accessed. The PA bus must be valid while PCS* is low. During DMA cycles, the PA address bus is ignored. During DMA cycles, only the FIFO data register is accessible.

PA16 low identifies an access to shared RAM. PA16 high identifies access to one of the ENI registers. Refer to section *6.6 ENI Host Interface* for a detailed description of the registers available in the ENI Interface.

When DMA FIFO operations are enabled, the PA15 input is used for the DMA acknowledge input PDACK*. DMA FIFO operations are enabled when the DMAE* bit in the ENI control register is set to 0.

When DMA FIFO operations are enabled, the PA14 output is used for the active low outbound FIFO DMA ready output PDRQO*. The PDRQO* signal is only routed through PA14 when both DMAE* and DMAE2 in the ENI control register are both set to 0. When DMAE* is set low and DMAE2 is set high, an active high version of

| Return/ Exception | Return Instruction | Previous State ARM R14_x | Previous State THUMB R14_x | Notes |
|---|---|---|---|---|
| BL | MOV PC, R14 | PC + 4 | PC + 2 | 1 |
| RESET | NA | — | — | 4 |
| UNDEF | MOVS PC, R14_und | PC + 4 | PC + 2 | 1 |
| SWI | MOVS PC, R14_svc | PC + 4 | PC + 2 | 1 |
| ABORT P | SUBS PC, R14_abt, #4 | PC + 4 | PC + 4 | 1 |
| ABORT D | SUBS PC, R14_abt, #8 | PC + 8 | PC + 8 | 3 |
| IRQ | SUBS PC, R14_irq, #4 | PC + 4 | PC + 4 | 2 |
| FIRQ | SUBS PC, R14_firq, #4 | PC + 4 | PC + 4 | 2 |

**Table 3-3: Exception Entry / Exit**

**Notes:**

1. Where PC is the address of the BL/SWI/Undefined Instruction fetch that had the prefetch abort.
2. Where PC is the address of the instruction that did not get executed since the FIRQ or IRQ took priority.
3. Where PC is the address of the Load or Store instruction that generated the data abort.
4. The value saved in R14_svc upon reset is unpredictable.

## 3.3.5 Reset Exception

When the ARM7TDMI is held in reset, the ARM7TDMI abandons the executing instruction and then continues to fetch instructions from incrementing word addresses.

When the ARM7TDMI is removed from reset, the ARM7TDMI:

1. Overwrites R14_svc and SPSR_svc by copying the current values of the PC and CPSR into them. The value of the saved PC and SPSR is not defined.
2. Forces the CPSR M field to 10011 (Supervisor mode), sets the I and F bits in the CPSR, and clears the CPSR T bit (back to ARM mode).

used as a simple block of RAM. Each cache-set can independently be configured as a 1K-byte block of cache or a 2K-byte block of RAM.

The NET+40 chip supports two cache control registers. Each control register identifies a configurable block of physical address space. The cache control register identifies all the cache and buffer features for the identified memory block. Typically, one cache control register identifies cacheable instructions while the other cache control register identifies cacheable data.

The cache tags are configured to identify individual 8-bit entries. This configuration maximizes the efficiency of a low-cost 16-bit bus running ARM Thumb code by minimizing the number of unused memory accesses. Traditional cache architectures require an entire cache-line (16 bytes per line) to be filled at a time. The traditional architecture approach can result in many unnecessary memory cycles when the program flow alters direction.

Each line in the cache RAM contains a 22-bit tag, 2 control bits, an 8-bit status field, and a 32-bit data field. Figure 3-3 provides a block diagram of the cache structure.

**TAG Tag Reference**

The TAG field identifies which memory location is currently stored within this cache entry. The TAG field is only valid when one of the four "V" valid bits are set to 1. The following formula can be used to calculate the actual physical address of the information for this cache entry.

physicalAddress =

```
(TAG << 10) || ((cacheAddress & 0x000007F8) >> 1)
```

**INVALID Invalidate**

0 - Location is value for cacheable entries

1- Location is invalid for any cache entries

The INVALID bit is used to prevent specific cache lines within a SET from being used for cache entries. The INVALID field is typically used to reserve cache entries for code that will be locked down in the cache at a later time.

**LOCK Locked State**

0 - Entry not locked

1- Entry locked in cache

The LOCK field is used to cause an entry to remain static within the cache. The LOCK field can be set to 1 in order to fix the current entry to remain in the cache SET forever. The LOCK bit is typically used to preload software functions in the cache.

**V3 Byte 3 Valid**

0 – Byte 3 is invalid

1 – Byte 3 is valid

The V3 bit is set to 1 to indicate that Byte 3 in this cache entry is valid. Byte 3 refers to the least significant address byte of a 32-bit word when operating in Big Endian mode, the most significant byte when operating in Little Endian mode.

**V2 Byte 2 Valid**

0 – Byte 2 is invalid

1 – Byte 2 is valid

The V2 bit is set to 1 to indicate that Byte 2 in this cache entry is valid.

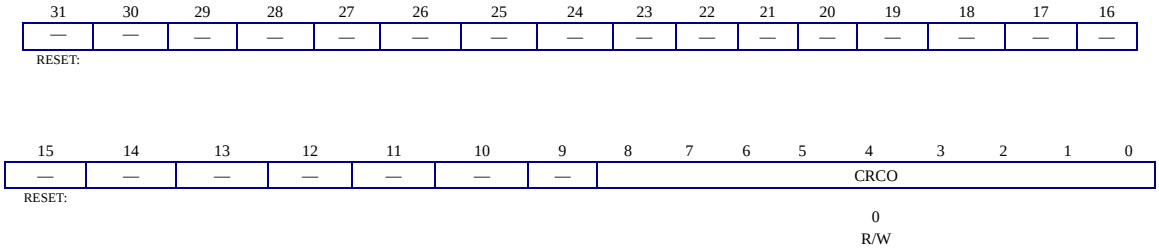
**V1 Byte 1 Valid**

0 – Byte 1 is invalid

## 5.3.11 STL Test Register

The STL test register is a 32-bit register that is only used during hardware simulation and never under normal operation conditions. All bits initialize to 0 on reset.

Address = FF80 0414




### CRCO

The CRCO field is only used for simulation of the STL module. This field must be set to 0 for normal functional operation of the STL Module.

### 6.2.3 IEEE 1284 Mode Configuration

Table 6-4 identifies the various modes for which each 1284 port can be configured. The configuration mode is provided using configuration bits in the 1284 port control registers. The proper operation mode is determined by the results of the IEEE 1284 negotiation process. The IEEE 1284 negotiation process must be done in firmware.

The NET+ARM chip provides hardware DMA support while operating in forward compatibility mode, forward ECP mode, and reverse ECP mode.

| Mode | EPP | ECP | BIDIR | MAN |
|---|---|---|---|---|
| Manual Forward Compatibility | 0 | 0 | 0 | 1 |
| Automatic Forward Compatibility | 0 | 0 | 0 | 0 |
| Nibble Mode | 0 | 0 | 0 | 0 |
| Byte Mode | 0 | 0 | 1 | 1 |
| Forward ECP | 0 | 1 | 0 | 0 |
| Reverse ECP | 0 | 1 | 1 | 0 |
| EPP | 1 | 0 | 1 | 0 |

**Table 6-4: IEEE 1284 Mode Configuration**

## 6.2.7 IEEE 1284 Byte Mode

A second method for obtaining reverse channel data for printers and peripherals is via the byte mode cycle. The byte mode cycle offers an enhancement over the nibble mode cycle since an entire byte can be transferred during a single cycle. The disadvantage to byte mode is that it cannot proceed while forward channel compatibility mode is in process.

Figure 6-7 describes the byte mode cycle. The AUTOFD* and STROBE* signals are manually manipulated.

1. The host signals ability to take data by asserting HostBusy low (AUTOFD*).
2. The peripheral responds by placing first byte on data lines.
3. The peripheral signals valid byte by asserting PtrClk low (ACK*).
4. The host sets HostBusy high to indicate that it has received the byte and is not ready for another byte yet.
5. The peripheral sets PtrClk high to acknowledge the host.
6. The host pulses HostClk (STROBE* via MSTB*) as an acknowledgment to the peripheral device.
7. States 1 through 5 repeat for additional bytes.

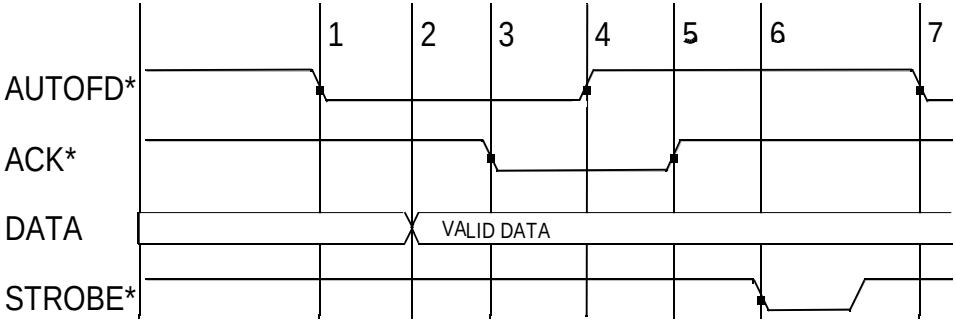



**Figure 6-7: Byte Mode Cycle**

The transition between forward compatibility mode and byte mode requires software intervention. When transitioning between forward compatibility and byte mode, the firmware must set the AUTOFD* (HostBusy) signal to 0 and the BIDIR signal to 1.

The transition between byte mode and forward compatibility mode requires software intervention. When transitioning between byte mode and forward compatibility mode, the firmware must wait for ACK* to be set high by the peripheral. After ACK* is set to 1, the firmware can set the BIDIR signal to 0. After the BIDIR signal is set to 0, the hardware immediately begins managing the forward channel data traffic.

**PINT1*** **ENI Interrupt 1**

The PINT1* signal is driven active low to indicate an interrupt condition to the external ENI processor. An interrupt condition is established when the ARM processor sets the STSINT or VDAINT bits in the ENI shared register. An interrupt condition is also established based upon the FIFO status and the FIFO interrupt enable bits in the FIFO mode mask/status register.

The PINT1* signal is further qualified by the EHWINT and SINTP2 bits in the ENI shared register. The ENI shared register is available in the ENI address space. The EHWINT provides global interrupt enable/disable control. The SINTP2 bit controls whether the interrupt is driven out the PINT1* pin or the PINT2* pin. The PINT1* is only driven active low when an interrupt condition is pending, the EHWINT bit is set to 1, and the SINTP2 bit is set to 0.

**PINT2*** **ENI Interrupt 2**

The PINT2* signal is driven active low to indicate an interrupt condition to the external ENI processor. An interrupt condition is established when the ARM processor sets the STSINT or VDAINT bits in the ENI shared register. An interrupt condition is also established based upon the FIFO status and the FIFO interrupt enable bits in the FIFO mode mask/status register.

The PINT2* signal is further qualified by the EHWINT and SINTP2 bits in the ENI shared register. The ENI shared register is available in the ENI address space. The EHWINT provides global interrupt enable/disable control. The SINTP2 bit controls whether the interrupt is driven out the PINT1* pin or the PINT2* pin. The PINT2* is only driven active low when an interrupt condition is pending, the EHWINT bit is set to 1, and the SINTP2 bit is set to 1.

The PINT2* signal can also be used to generate an interrupt from the external ENI Processor to the ARM processor. The PINT2* pin becomes an input when the DINT2* bit is set to 0 in the ENI control register. The DINT2* bit can be automatically initialized to its proper state upon reset by including or not including a pull down resistor on the A5 signal (refer to Section *6.5.1 ENI Module Hardware Initialization*). When DINT2* is configured as in the pulsed interrupt input, the low to high transition on the PINT2* input causes an interrupt condition to be established to the ARM processor. The low to high transition on the PINT2* input causes the INTIOF bit to be set in the shared register for the ARM processor.

The PINT2 signal can also be used to provide an active high DMA request when the ENI interface is configured to operate in FIFO mode and the DMAE2 bit is set in the ENI control register. In this condition, the inbound and outbound DMA ready signals are logically “ORed” together and routed through the PINT2 pin as an active high DMA request. When the DMAE2 bit is set, the PA14 and PA13 signals are no longer used as DMA request outputs and return to their shared RAM address function extending the amount of addressable shared RAM to 32K while operating in FIFO DMA mode.

**PBRW*** **ENI Data Buffer Read/Write**

The PBRW* signal is an output used to control the data direction pin for an external data bus transceiver. Some applications require a data bus buffer between the NET+ARM chip and the external ENI processor system. PBRW* high indicates a data transfer from the NET+ARM chip to the ENI processor; while PBRW* low indicates a data transfer from the ENI processor to the NET+ARM chip.

**PEN*** **ENI Data Buffer Enable**

The PEN* signal is an output used to control the output enable pin for an external data bus transceiver. Some applications require a data bus buffer between the NET+ARM chip and the external ENI processor system. PEN* low indicates a data transfer between the NET+ARM chip and the external ENI processor is in progress.

## 7.3 General Purpose I/O Configurations

The GEN module provides the physical layer connections for NMSI (Non

When the PORTB MODE bit is set to 1 and the PORTB DIR bit is set to 0, the PORTB bit is configured to operate in Special Function Input Mode. When the PORTB MODE bit is set to 1 and the PORTB DIR bit is also set to 1, the PORTB bit is configured to operate in Special Function Output Mode.

| PORTB BIT | GPIO Mode | | Special Function Mode | |
|---|---|---|---|---|
| | BMODE=0 | | BMODE = 1 | |
| | BDIR=0 | BDIR=1 | BDIR=0 | BDIR=1 |
| PORTB7 | GPIO IN | GPIO OUT | | TXDB |
| PORTB6 | GPIO IN | GPIO OUT | DREQ2* | DTRB* |
| PORTB5 | GPIO IN | GPIO OUT | | RTSB* |
| PORTB4 | GPIO IN | GPIO OUT | SPI-S-CLK-IN-B*<br>RXCB-IN | SPI-M-ENABLE-B*<br>RXCB-OUT<br>OUT1B* |
| PORTB3 | GPIO IN | GPIO OUT | RXDB | |
| PORTB2 | GPIO IN | GPIO OUT | DSRB* | DACK2* |
| PORTB1 | GPIO IN | GPIO OUT | CTSB* | |
| PORTB0 | GPIO IN | GPIO OUT | DCDB*<br>DONE-IN2* | DONE-OUT2* |

**Table 8-4: PORT B Configuration**

**PORTB7**

The PORTB7 bit can be configured for GPIO Input mode or GPIO Output Mode.

The PORTB7 bit can be configured for Special Function Output. When configured for Special Function Output, the PORTB7 signal provides the Transmit Data (TXD) signal for Serial PORT B. The TXD signal is used for Serial Transmit Data when configured to operate in UART, SPI, or HDLC modes.

The PORTB7 bit has no useful function when configured to operate in Special Function Input mode.

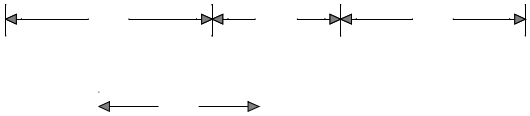


**Figure 13-23: ENI Single Direction DMA Timing**
Refer to Table 13-13 for values.

| Num | Characteristic | Min | Max | Unit |
|---|---|---|---|---|
| 100 | TCK low to TDO valid | 0 | 50 | ns |
| 101 | TCLK low to TDO high impedance | 0 | 20 | ns |
| 102a | TDI, TMS valid to TCK high (setup) | 50 | | ns |
| 102b | TCK high to TDI, TMS invalid (hold) | 50 | | ns |
| 103 | TCK Cycle Time | 100 | | ns |
| 104 | TCK pulse width | 40 | | ns |
| 105 | TRST* low time | 100 | | ns |
| 106 | TRST* valid to TCK low (setup) | 40 | | ns |

**Table 13-15: JTAG Timing**

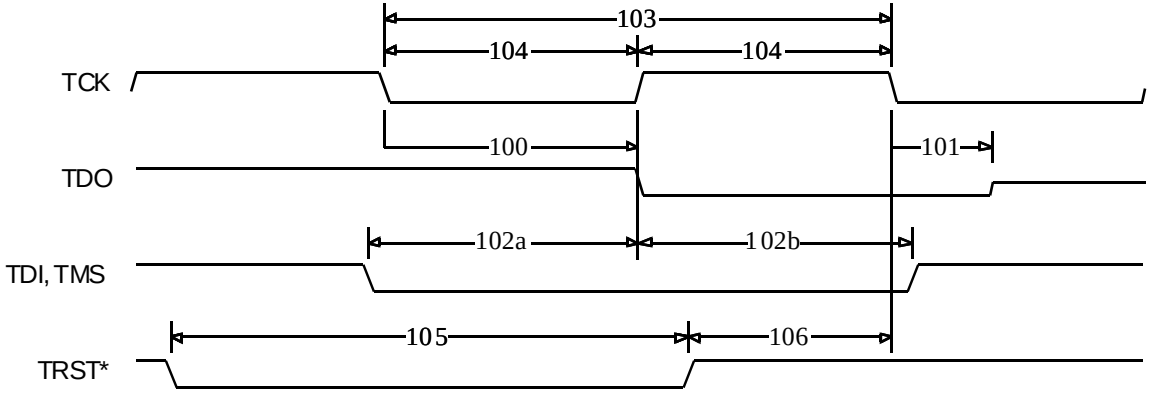



**Figure 13-28: JTAG Timing**
Refer to Table 13-15 for values.

# Index

## Numerics



## A



## B