Product Brief
netX10

next Generation of Communication Controllers

Language: English

Preliminary

www.hilscher.com

1 Introduction

This document provides an overview on the netX10 network controller, which combines the communication power of netX core technology with highly integrated control peripherals in a small package to allow sophisticated single chip solutions for next generation networked devices in all kinds of sensor/actuator industrial applications.

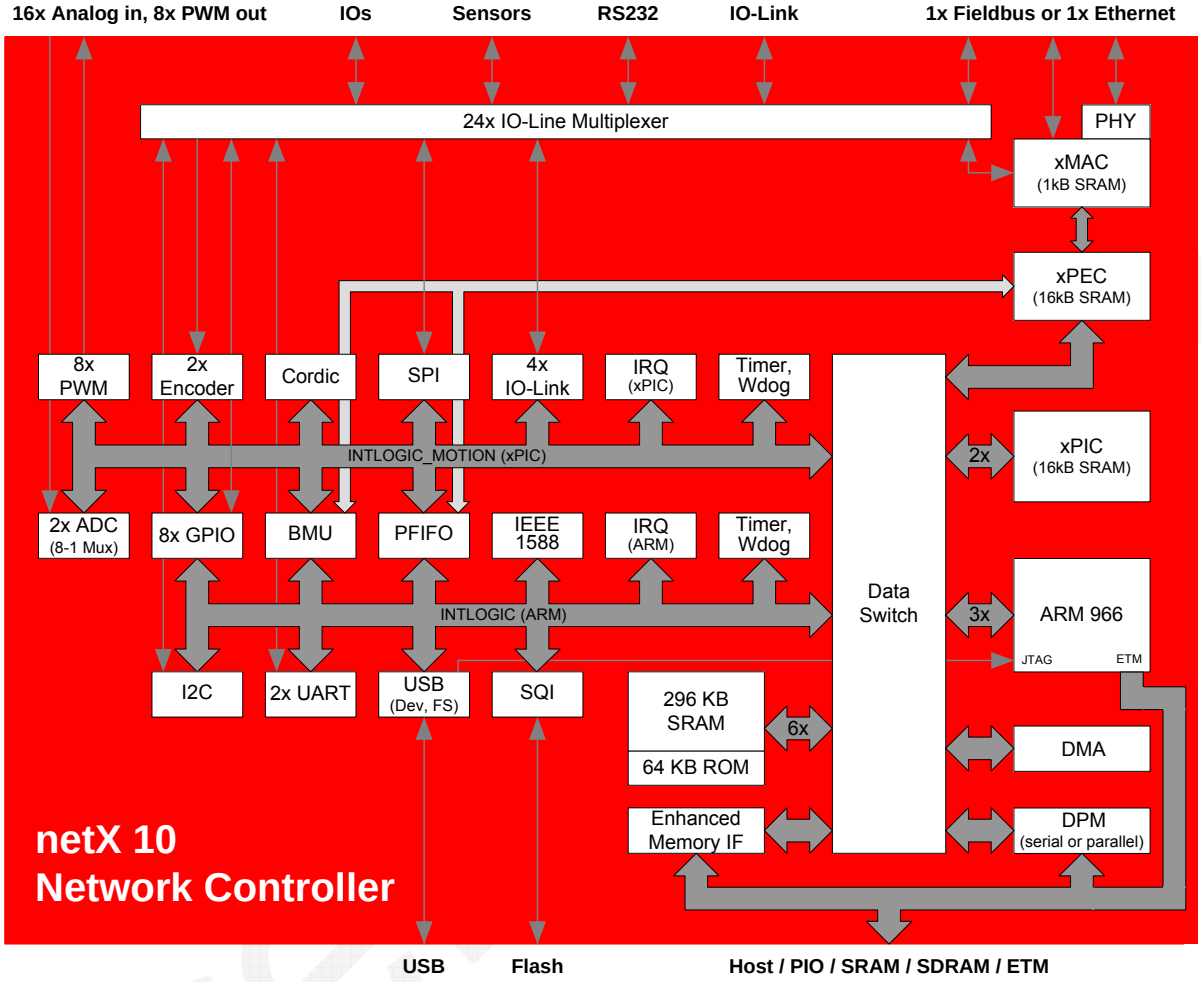


Fig. 1: netX10 Block Diagram

The basic functionality of netX10 is derived from the netX50 network controller architecture. The netX10 controller provides a single programmable communication channel supporting all kinds of fieldbus interfaces, an IEEE1588 Time unit as well as an integrated PHY to support a variety of non-switched Ethernet networks.
As all members of the netX controller family, the netX10 benefits from the already well-proven xMAC/xPEC communication channel technology, where the Medium-Access-Controller (xMAC) sends and receives the serial data streams according to the respective bus access process and converts them from Bit to Byte streams, while the Protocol Execution Controller (xPEC) compiles the bytes delivered by the xMAC into data packets and controls the telegram traffic. This allows the implementation of the most varied protocols which can be synchronized independently of the Host CPU response time.
A large number of additional functional units have been integrated to provide high flexibility for different applications. In order to support applications like fast IO or fast and precise sensor or actuator interfaces, a dedicated, user programmable xPIC processor, highly optimized for fast response and high calculation power, has been added. Along with the high resolution PWM and Encoder Units and a dedicated MATH Function Accelerator, advanced motion control or signal processing applications can easily be realized with a single netX10 chip.

1.1 Product Features

True Multi Core System

- 100 MHz internal clock (10 ns Cycle Time)
- ARM966 core with optimized bus connection and ETM cell
- High performance Data-Switch
- xPEC/xMAC channel for communication protocols
- dedicated xPIC core for applications
- dedicated environment like Vector Interrupt Controllers, Timers, Watchdogs for ARM and xPIC

Large On-Chip Memory

- 296 KByte SRAM (no external RAM required for most applications) comprising 6 separate segments (4 * 64K, 1 * 32K, 1 * 8K) for fast parallel access
- 32 KByte local memory for xPEC and xPIC
- 64 KByte ROM

Host and Memory Interface

- Enhanced 8/16 Bit Memory Controller, supporting SDRAM, FLASH (page/burst mode) and READY signal input for external wait state generation
- Configurable Dual Port Memory Interface (8/16 Bit or fast SPI Slave (80MHz))
- Fast SQI Controller (nibble SPI) with up to 200Mbit/s throughput (no need for parallel flash)
- Dynamic data/code upload with SQI and DMA

xPIC – flexible Peripheral Interface Controller for User Applications:

- 32-bit RISC CPU, optimized for fast I/O processing (e.g. motor control applications)
- Instruction Set extended by DSP commands for control and signal processing applications
- Fast interrupt response due to Dual Register Bank
- Local memory for fastest calculation/response (8 Kbyte Instruction/ 8 KByte Data) and local system bus to motion control units
- Low Overhead Loops
- Harvard Architecture 32 Bit Instruction / 32 Bit Data Bus
- 32-bit linear address range (local, on-chip, external memory access)
- 32x32->64 Hardware Single Cycle Multiplier
- All arithmetic functions signed/unsigned, with/without saturation and with conditional execution (if register based)
- Single cycle ALU operations with parallel Register/Memory Transfer
- 16 work registers
- Interrupt Controller (Priority based and Fast Interrupt support)
- 32-bit Timer/Counter
- Hardbreaker – hardware debug unit (single step, 2 HW breakpoints, SW breakpoints)
- Programmable in C/C++/Assembler. Eclipse based tool chain for development and debug
- Support of netX10 as Matlab/Simulink Embedded Target for rapid function evaluation and prototyping is under development

Sensor/ Actuator/ Motion Control Features

- Enhanced PWM Unit (2,5 ns resolution, 8 channels, dual time base, sync and IRQ support)
- 2x Enhanced Encoder Unit (with filter, capture, sync, IRQ and additional support for precise velocity measurement)
- 2 Integrated 10-bit ADCs with 8 input channels each, 1 MSample/second, programmable sequencer unit with trigger and interrupts for enhanced synchronization
- Enhanced Interrupt/ Synchronization Support for all units
- CORDIC based MATH function accelerator accessible by all internal processors, primarily used for converting polar to Cartesian coordinates with motor control applications.

Peripherals

- IEEE1588 System Time Unit
- Integrated 10/100Mbit Ethernet PHY
- 1 x USB device full speed
- 2 x UART (16550 compatible, with FIFO)
- 1 x SPI (up to 50 MHz), Master and Slave
- 1 x I2C (up to 3.4 Mbit/s)
- 8 x GPIO with enhanced capture and PWM capabilities
- 24 Line Multiplex Matrix for flexible pin function switching with PIO functionality
- Up to 47 additional PIOs (unused pins of host / memory interface)
- Watchdog Timer

Proven Process:

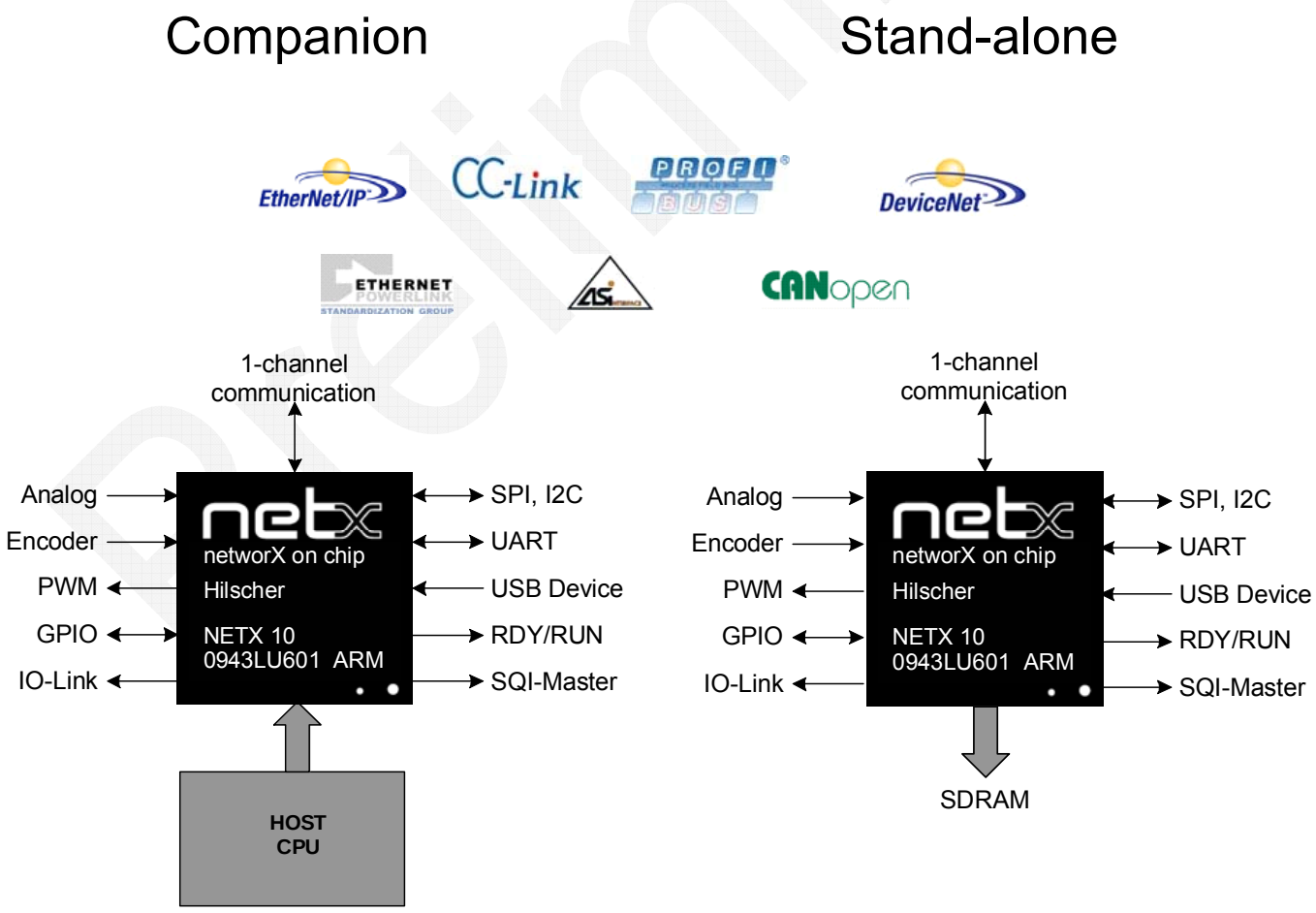
- NEC 150 nm technology
- over 10 years chip availability
- -40..+85°C

1.2 Typical Applications

The variety of sensors, IO modules and actuators represent a large segment of the automation market. Well known examples of applications are block IO modules, valve blocks, encoder, identification systems (RFID), simple process controllers or PLCs, light barriers and switches, text displays, electronic scales etc. Most of these devices need to be connected to higher level control systems via fieldbus interfaces and increasingly via Ethernet network (standard TCP/IP communication, ModbusTCP or Ethernet/IP).
While most current solutions consist of a main microcontroller, accompanied by an additional communication controller as well as various peripherals and glue logic, the cost sensitive nature of such products actually demands a highly integrated single chip solution, which ideally should provide the flexibility of an FPGA-based solution.
Unlike with complex solutions in the drive or control engineering, only a narrowed functionality and data set is required here, resulting in substantially reduced requirements for processing power and memory size.
With its considerably large amount of integrated memory, integrated peripherals and the processing power of the ARM CPU and the xPIC core, combined with its freely programmable communication channel, the netX10 perfectly suits this field of applications.

Applications Summary

- Single Chip IO, sensor, actuator, motion or process controller with communication interface
- Designed to support all Fieldbus (Profibus, CAN, DeviceNet, CCLink etc.)
- or Real Time – Ethernet (ModbusTCP, Ethernet IP, Powerlink etc.)



2 Functional Overview

2.1 Multi Core System

The netX10 is a true multi processor system on chip. The computing power of netX10 is composed of an ARM CPU, an xPIC for user applications and an xPEC/xMAC unit for communication protocols. Due to the parallel architecture and efficient communication between processors the overall computing performance of the netX10 may reach 1900 MIPS, while realistic average performance on common applications including communication and control tasks reaches at least 900 MIPS.

2.1.1 ARM 966

The main processor of the netX10 is an ARM966 core running on 100MHz clock, a well known standard CPU for general purpose and embedded applications. The connection of the ARM966 core to the data switch has a total of three ports in order to achieve higher data throughput and lower wait states (1.5 x performance of netX50 ARM, though half the system clock frequency) The ARM core in netX10 is mainly used for processing of higher levels of communication protocols, while still leaving processing power for user specific applications.

2.1.2 xPEC/xMAC

The xMAC core together with the xPEC core represent one xC channel (communication channel), optimized for lower levels of communication protocols, allowing fast and deterministic response to real-time events and combining the flexibility and performance of FPGA based solutions with the reliability and low unit cost of an ASIC design. With the xC cores running protocol specific microcode on assembler language level, virtually any serial communication protocol can be realized with an xC channel.
The instructions for the RPU (Receive Processing Unit) and TPU (Transmit Processing Unit) are 64 Bit wide and include up to eight commands per instruction, being all executed simultaneously. Theoretically this yields a computing performance of 1700 MIPS. With common applications, RPU and TPU programs utilize two to three commands per instructions on average, resulting in a performance of ca. 600 MIPS.

2.1.3 xPIC

The xPIC is a flexible Peripheral Interface Controller designed especially for user applications. The xPIC is a general purpose 32-bit RISC CPU with an instruction set optimized (DSP extensions) for fast and deterministic data processing as needed in many real industrial applications. The xPIC can be used as a fast CPU to filter, analyse, collect, convert and process sensor data, ranging from simple IOs to complex encoders and sensors with an analog front-end. Another range of applications is the control of any actuators ranging from simple digital and analog IOs, pumps, valves, or switches to the control of virtually any type of electric motor. Along with the integrated encoder, PWM and CORDIC units, the xPIC can be used to realize many kinds of control applications in the motion or process control field.

2.2 Internal Memory

To allow true single chip solutions, the netX10 is equipped with 296 KByte internal SRAM, which comprises 6 separate blocks, each with its own connection to the main data switch. This allows simultaneous high performance access to memory by ARM, xPIC, xPEC, DMA controller and DPM interface.
The size of netX10 internal memory was chosen to allow the processing of any fieldbus and Ethernet protocol without the need for an external RAM. However an efficient Memory controller, allowing the use of external SDRAM and FLASH is still available to support all kinds of memory intensive applications.

2.3 Data Switch

The well proven Data Switch concept that can already be found in preceding members of the netX family and which ensures independent and parallel internal data exchange, has also been implemented in the netX10. However the netX10 data switch has been enhanced with a second Slave I/O bus, allowing both, ARM CPU and xPIC, to access internal units through a separate bus concurrently.

While the xPIC Intlogic bus hosts units which are likely to be mainly accessed by the xPIC, the ARM Intlogic bus provides access to units which will usually be accessed by the ARM CPU. Both buses and hence all units can however be reached by both processors and while most units are connected to only one of these buses, the GPIO, the Buffer Manager and the Pointer FIFO even have a path to both buses.

2.4 DMA Controller

The netX10 is equipped with a versatile 3 channel DMA-controller, allowing all kinds of data transfers between netX10 units.

2.5 Dual Port Memory Interface

The Dual Port Memory interface is commonly used to connect the netX10 to a user application specific host system. In order to make all of the netX functionality accessible to the host CPU, the different blocks of the Dual Port Memory interface can be mapped freely within the internal memory area.

The netX10 DPM interface can either operate as a parallel (8 or 16 Bit wide) DPM interface or as a high speed serial DPM interface based on standard SPI. In parallel mode, a standard SRAM like access protocol and a multiplexed protocol are (e.g. for interfacing to TI OPMAP controllers) supported.
Being a completely new developed module, the DPM timing could be optimized for high data rates, ensuring burst access times of down to 40 ns and random access times down to 60 ns. Enhanced communication features (e.g. handshake cells) of the DPM interface known from netX50/100/500 have also been implemented. This provides (host-) software compatibility, allowing to replace existing netX50 based systems by netX10 systems where one communication channel is sufficient. Two optional widely configurable interrupt signals are also available as also known from netX50/100/500.

As with all netX host interfaces, also the netX10 allows using any spare signals (e.g. upper data lines in 8 Bit mode or unused address lines) as PIO signals.
Since the small package of the netX10 requires sharing pins among the DPM interface and the Memory Controller, the concurrent use of external memory and DPM interface is only possible when operating the DPM interface in serial mode. The following table provides an overview on the possible DPM / Memory modes:

Mode	Notes
Parallel DPM (128K) 16 bit + 3 PIOs	DPM size is configurable to 2K, 8K, 16K, 32K, 64K and 128K. Using a reduced DPM size and / or width, frees up a corresponding number of address lines, which can then be used as PIOs.
Parallel DPM (128K) 8 bit + 11 PIOs	
Parallel DPM (2K) 16 bit + 9 PIOs	
Parallel DPM (2K) 8 bit + 17 PIOs	
Serial DPM + 41 PIOs	4 SPI signals + 2 IRQ signals
Serial DPM + 64MB SDRAM 8bit + 16MB SRAM/FLASH 8bit	The memory controller can address up to 512 MB SDRAM, however the largest available components are currently 512 Mbit types (8 Bit types: 64Mx8 = 64 MB)
64(128)MB SDRAM 16bit + 16MB SRAM/FLASH 16bit	16 Bit types: 32Mx16 128 MB configuration requires two 512Mx8 SDRAM devices
+ 64MB SDRAM 8bit + 16MB SRAM/FLASH 8bit + 9 PIOs	In memory mode, unused signal lines (upper data lines in 8 Bit mode) can also be used as PIOs

Table 1 – DPM / memory modes

2.6 Communication Controller

The Communication Controller inside netX10 is composed of a flexible xC channel, processing the lower communication protocol layers and the ARM 966 CPU, taking care of the upper protocol layers. The following subchapters list the common applications for the communication controller.

2.6.1 Standard Ethernet

With the appropriate microcode (provided by Hilscher), the xMAC processor of the netX10 can provide standard Ethernet functionality like handling of Ethernet frames, CRC32 check, etc. Together with the xPEC processor, more complex functions like frame filtering or intelligent frame based DMA are possible, allowing to easily realize protocols like Open Modbus TCP or EtherNet/IP.
The integrated Ethernet PHY reduces external circuitry for the Ethernet port to termination resistors and a transformer.

2.6.2 Real Time Ethernet

Many Real Time Ethernet protocols require certain hardware features which can not be provided by standard Ethernet MACs that can be found in many communication controllers. Using netX devices, these special features are realized on xMAC/xPEC processors by the appropriate microcode provided by Hilscher.
Since most Real Time Ethernet applications require at least two Ethernet ports, which also requires two xC channels on netX technology, the netX10, providing a single xC channel only, can not support protocols like PROFINET RT/IRT, EtherCAT or SERCOS III. However, along with an external HUB, Ethernet Powerlink applications are still possible. In that case, the xPEC responds to a Powerlink Poll-Request-Telegram immediately without involving the ARM CPU.
For synchronisation purposes, the netX10 also contains an IEEE 1588 time unit.

2.6.3 Fieldbus Interface

Fieldbus interfaces usually require only one communication channel (xC unit) and can hence be fully supported by netX10. With the corresponding microcode running on xMAC/xPEC processors, virtually any serial communication protocol can be supported, like:

- AS interface Master
- CANopen Master or Slave
- CC-Link Slave
- DeviceNet Master or Slave
- PROFIBUS-DP Master or Slave

2.7 Memory Controller

The netX10 is equipped with an enhanced 8/16 Bit memory controller, supporting SDRAM, SRAM, parallel FLASH (including burst accesses with fast asynchronous page mode FLASHes) and an external wait signal, allowing to connect devices with a non deterministic access time.

2.8 SQI Interface

The standard SPI0 controller known from the netX50 has been enhanced with an additional “quad mode” supporting 1-bit standard SPI devices as well as 4-bit SQI devices. In 4-bit mode a maximum data transfer rate of 200 Mbit/s is possible. Together with the DMA controller SQI can be used to load code or data dynamically at runtime (web-pages, optional services etc.). The SQI mode is already available during the boot phase, allowing fast system start-up.

3 Special Functions

3.1 xPIC – Peripheral Interface Controller

One of the new features invented with the netX10 is the xPIC, a special processor, optimized to control different kinds of sensors and actuators allowing fast response and data processing.

3.1.1 General Structure

The xPIC core is a generic 32-Bit RISC processor with Harvard architecture. To ensure maximal calculation power and fast deterministic response the xPIC core has its own local SRAM, consisting of 8 KByte instruction and 8 KByte data memory.
Figure 2 shows the internal structure of the xPIC processor. The Arithmetic Logical Unit (ALU) and Address Unit together with 13 Working Registers are the “heart” of the xPIC processor, which has a 32-bit instruction and a 32-bit data bus resulting in a 32-bit address space, allowing the xPIC to access the complete netX10 internal memory range.

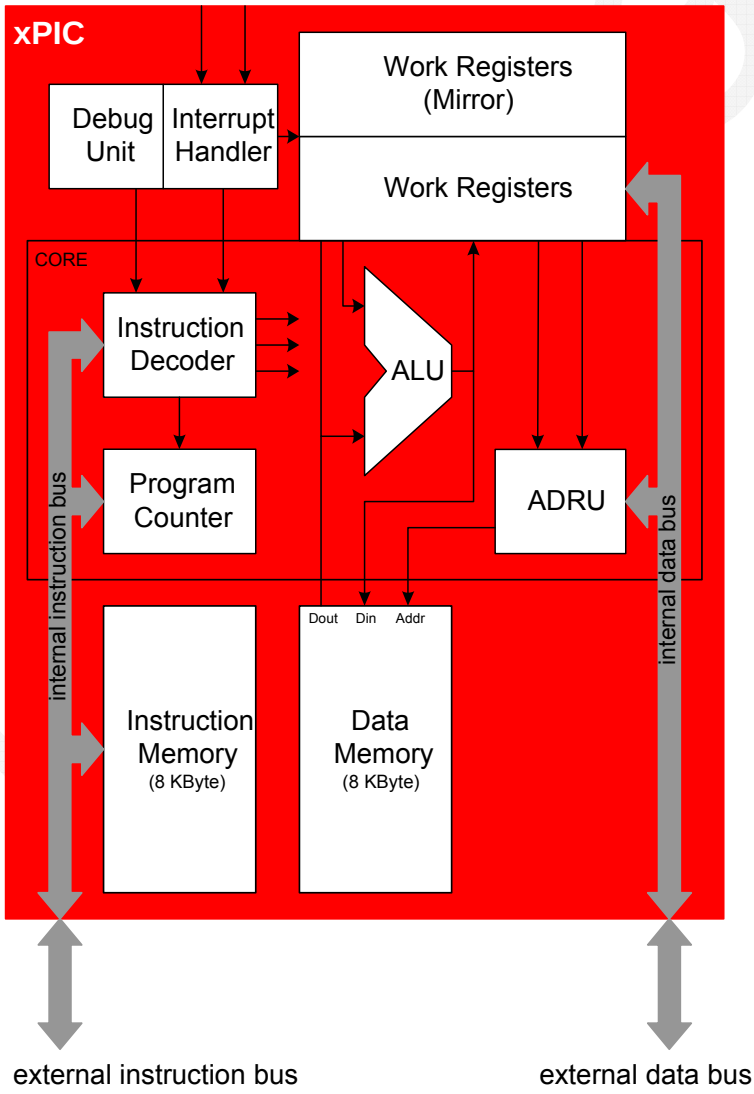


Fig. 2: Internal structure of xPIC core

A 3-step pipeline ensures fast response time and the optimized core logic allows the execution of ALU instructions in parallel to register transfers, which results in more compact code and a processing power of more than 100MIPS.

3.1.2 Debug Unit

To allow efficient debugging of the program code, the xPIC contains a Hardbreaker debug, providing the following features:

- Single Stepping
- Two Hardware Breakpoints
- Software Breakpoints
- Real-Time Watch mode

The functions of the debugging unit are accessible through registers in the netX10 address space.

3.1.3 Interrupt Controller

To allow fast interrupt response, the xPIC is backed by its own Interrupt Controller which can generate fast interrupts (FIRQ) as well as normal interrupts (IRQ). The incoming interrupt requests from any netX10 unit can be configured as FIRQ or IRQ. A second (mirrored) register bank allows the switching to FIRQ code within 50 ns. A special “return_firq” instruction ensures returning from the FIRQ in the same way. Normal interrupts (IRQ) can be configured with the required priority.

3.1.4 Instruction Set

The instruction set of the xPIC processor is optimized for high performance data processing. Beside the standard ALU instructions for signed and unsigned addition, subtraction, arithmetical and logical shift, some special instructions have been added to allow more efficient data processing and control algorithms:

- SAT Signed and unsigned saturation is supported for all arithmetical instructions.
- MAX/MIN - maximal/minimal of two values is supported for signed and unsigned numbers. These are useful for efficient signal limitation or sorting routines.
- ARITHM – single cycle signed arithmetical mean with round to nearest even.
- JUMP with auto decrement in single cycle makes low overhead loops possible.
- CLZ/CLO/CLMSB – Counts leading zeros/ones/MSBs. Important for efficient normalization, data type conversion and support of floating point numbers
- MUL – signed/unsigned Hardware Multiplier provides 32x32->64 single cycle multiply
- IF(cond) – conditional execution of ALU instructions.
- Extended Status/Conditions Flag register

3.2 Fast IO and Motion Control Functions

As these units will typically be controlled by the xPIC, they are internally connected to the xPIC Intlogic Bus, however they are also accessible by the ARM CPU if necessary.

3.2.1 PWM Unit

The netX10 is equipped with an optimized PWM unit with a high resolution of 2.5 ns. The PWM unit has two time base counters, which both can be used as a source for any of the eight compare units. The compare level can be set for every compare unit directly or at the next timer zero-crossing via shadow register.
The PWM signals are shared with Multiplex Matrix signals MMIO[11:4], while each of the eight PWM outputs can be enabled and configured individually. A configurable failure unit allows to use any of the 24 MMIO signals as failure input which can individually set the PWM outputs to their predefined state (low, high, Hi-Z or “don’t-touch”).
Two different Interrupts can be generated by the PWM unit on counter_zero, counter_max, and positive or negative signal edge on any PWM signal in input mode, while each IRQ can be delayed up to 20.48 us (programmable in 20 ns steps). This feature can be efficiently used for exact sampling and synchronization of any measurement device (e.g. ADCs for motor current measurement or encoder capture).

The following figure shows the internal structure of the PWM unit:

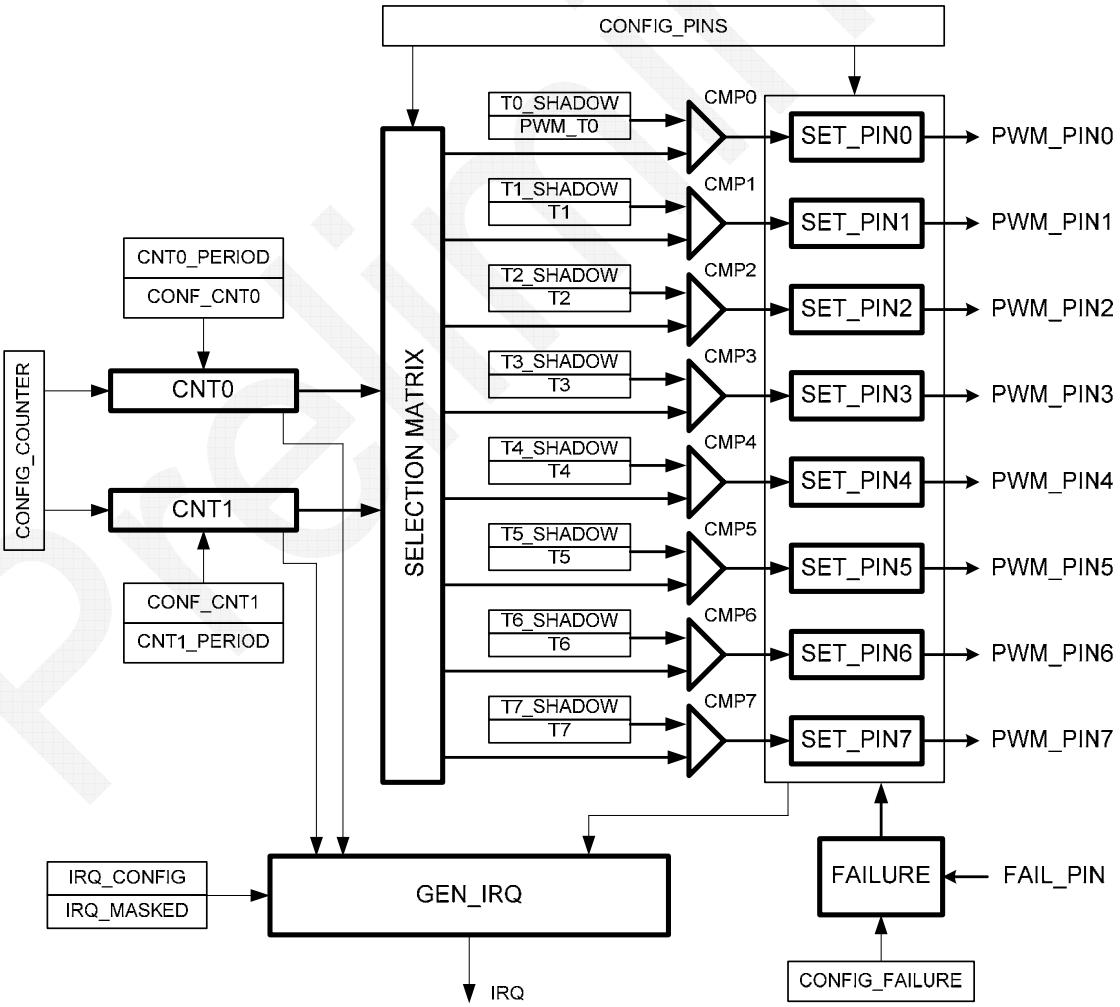


Fig. 3: Internal structure of the PWM unit

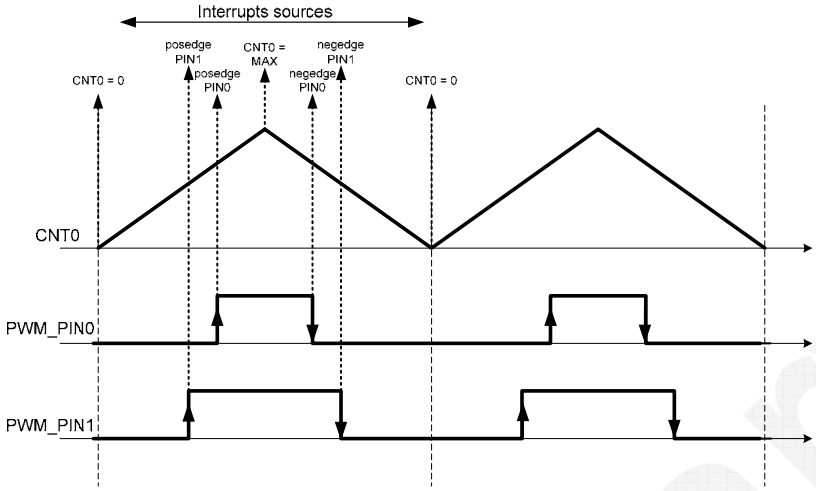


Fig. 4: Configurable interrupt sources of the PWM unit

The PWM unit can be configured to generate the waveforms to control most known motor types (DC motors, stepper motors, synchronous and asynchronous motors …) as well as to generate the precise excitation signals for resolvers. The counters of the PWM unit can capture the System Time (which is adjustable), allowing to synchronize the PWM waveforms of many devices in the network with a precision up to 10 ns.

3.2.2 Encoder Interface Unit

The netX10 includes two identical encoder units. Beside the general functionality to count the pulses of incremental encoders, the unit supports advanced time capture functions (4 different capture registers) and interrupt generation on 16 different encoder events. The sampling of the encoder values can simultaneously capture the current System Time for synchronization purposes. A special Ta/Te mode allows very precise determination of rotary speeds especially at low speeds.

3.2.3 ADC Unit

The netX10 includes two 10-bit AD Converters, each with an analog 8 input multiplexer. Two input channels can be sampled simultaneously at a rate of up to 1 MS/second. The channel selection for each following conversion can be programmed by a sequencer unit. The start of conversion can be precisely synchronized with other netX units (e.g. to PWM unit for motion control applications using resolvers). Completed conversion or ADC ready for start of next conversion events can generate an interrupt to ARM or xPIC. A special pipeline mode allows continuous sampling of analog values from different channels, as the next channel can be programmed for sampling during the current conversion.

3.3 CORDIC Unit

To improve the performance of the data processing and control algorithms of ARM or xPIC CPUs, the netX10 is equipped with a special CORDIC (CoOrdinate Rotation DIgital Computer) based hardware unit, which dramatically speeds up the calculation of transcendental functions such as sin, cos, e^x, lnx, div, sqrt, atan2, exp, asin, acos. The precision of the unit is configurable and due to optimized hardware logic the 32bit precision values can be calculated within just 20 cycles. Depending on the function, a total calculation time of 400 ns can be reached. The CORDIC unit supports rational, hyperbolic and linear mode while a dual table of coefficients, allows fast switching between the modes.

3.4 Peripheral Functions

3.4.1 General Purpose IO (GPIO)

The netX10 contains a GPIO module as already known from netX50/100/500, providing 3 hardware timers and 8 I/Os.

3.4.2 Programmable IO-Link Controller

The netX10 is equipped with four programmable serial XLINK controllers, connected to the Multiplex Matrix, which can, together with the xPIC, operate as I/O-Link Master Controllers.

3.4.3 UART

Two 16550 compatible UART units are integrated, which external signals (RX, TX, RTS, CTS) are accessible through the Multiplex Matrix. Internally, the UARTs are connected to the ARM Intlogic Bus.

3.4.4 SPI

A second dedicated SPI unit can be used as a fast IO interface to external sensors, high precision ADCs etc. Its signals are accessible through the Multiplex Matrix. Internally, this unit is connected to the xPIC Intlogic Bus.

3.4.5 I2C

The netX10 also provides an I2C compatible interface, allowing to connect many components with an I2C or “two-wire” interface (e.g. integrated temperature sensors).

3.4.6 Debug Interface

The netX10 is equipped with a standard JTAG interface which can be used for debugging or Boundary Scan Testing and further provides an ETM interface, which is shared with the DPM / Memory signals.

3.4.7 I2C for Secure Memory / Boot Options / System LED

As known from preceding members of the netX family, the netX10 provides two multipurpose I/Os (RDY and RUN) which are used for accessing the netX secure EEPROM, selecting boot options and driving the dual SYS LED, providing status information.

4 Development Tools

Besides full featured support of development tools for integrated ARM966 core Hilscher provides a sophisticated tool chain for their xPIC core.

4.1 Compiler/Assembler/Linker

Hilscher provides complete compiler/assembler/Linker tool chain for xPIC processor.

4.2 IDExPIC – Integrated deveplopment environment

Hilscher will provide an Eclipse based state of the art integrated development environment to create, edit and debug xPIC programs, which is currently under development. Please check for availability.

5 Device Pinning

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A		DPM_A16 MEMSR_A16 MEMDR_RASn PIO00 ETM_DACK	DPM_A14 MEM_A14 MEMDR_BA0 PIO54 ETM_TCLK	DPM_A12 MEM_A12 PIO52 ETM_TPKT03	DPM_A10 MEM_A10 PIO50 ETM_TPKT01	DPM_A08 MEM_A08 PIO48 ETM_PSTAT2	DPM_A06 MEM_A06 PIO46 ETM_PSTAT0	DPM_A04 MEM_A04 PIO44	DPM_A02 MEM_A02 PIO42	DPM_A00 MEM_A00 MEMSR_DQM0 PIO40	DPM_RDY MEMSR_RDY PIO61	MEMDR_CLK DPM_SIRQ PIO63	DPM_D15 MEM_D15 PIO15 ETM_TPKT15	DPM_D07 MEM_D07 PIO39 ETM_TPKT07	
B	DPM_A17 MEMSR_A17 MEMDR_CASn PIO01	DPM_A15 MEMSR_A15 MEMDR_BA1 PIO55 ETM_TSYNC	DPM_A13 MEM_A13 PIO53 ETM_DREQ	DPM_A11 MEM_A11 PIO51 ETM_TPKT02	DPM_A09 MEM_A09 PIO49 ETM_TPKT00	DPM_A07 MEM_A07 PIO47 ETM_PSTAT1	DPM_A05 MEM_A05 PIO45	DPM_A03 MEM_A03 PIO43	DPM_A01 MEM_A01 PIO41	DPM_RDn MEMSR_OEn MEMDR_CKE PIO58	DPM_WRn MEMSR_WRn MEMDR_WEn PIO59	DPM_CSn MEMSR_CSn PIO60	DPM_BHEn MEM_DQM1 PIO57	DPM_D14 MEM_D14 PIO14 ETM_TPKT14	DPM_D06 MEM_D06 PIO38 ETM_TPKT06
C	DPM_A18 MEMSR_A18 MEMDR_DQM0 PIO02	DPM_A19 MEMSR_A19 PIO03	*NC*	VSS	VDDIO	VSS	TEST	VSS	VDDIO	VSS	TMC1	VSS	VDDIO	DPM_D13 MEM_D13 PIO13 SDPM_SIRQ ETM_TPKT13	DPM_D05 MEM_D05 PIO37 ETM_TPKT05
D	DPM_A20 MEMSR_A20 PIO04	DPM_A21 MEMSR_A21 PIO05	VSS	*NC*	MMIO21	MMIO20	MMIO19	MMIO18	MMIO17	MMIO16	MMIO15	MMIO14	VSS	DPM_D12 MEM_D12 PIO12 SDPM_DIRQ ETM_TPKT12	DPM_D04 MEM_D04 PIO36 ETM_TPKT04
E	DPM_A22 MEMSR_A22 PIO06	DPM_A23 MEMSR_A23 PIO07	VDDIO	MMIO22	VDDC	*NC*	*NC*	VDDC	*NC*	VDDC	*NC*	MMIO13	TMC2	DPM_D11 MEM_D11 PIO11 SDPM_CLK	DPM_D03 MEM_D03 PIO35 ETM_TPKT11
F	OSC_XTI	JT_TDI	*NC*	MMIO23	*NC*	*NC*					VDDC	MMIO12	VSS	DPM_D10 MEM_D10 PIO10 SDPM_CSn	DPM_D02 MEM_D02 PIO34 ETM_TPKT10
G	OSC_XTO	JT_TDO	BIST_TRSTn	VSS	VDDC						*NC*	DPM_DIRQ MEMDR_CSn PIO62	VDDIO	DPM_D09 MEM_D09 PIO09 SDPM_MOSI	DPM_D01 MEM_D01 PIO33 ETM_TPKT09
H	RDY	RUN	OSC_VSS	JT_TMS	VSS						*NC*	VDDC	VSS	DPM_D08 MEM_D08 PIO08 SDPM_MISO	DPM_D00 MEM_D00 PIO32 ETM_TPKT08
J	USB_DNEG	VDDIO	OSC_VDDC	USB_VSS	USB_VDDIO						PHY_VSSAT	PHY_VDDIOAT	VDDC	VDDC	VSS
K	USB_DPOS	VSS	*NC*	JT_TCLK	*NC*						PHY_VSSAT1	PHY_VDDCART	PHY_VSSAR	PHY_RXP	PHY_RXN
L	PORn	SPI0_CS0n	VDDIO	JT_TRSTn	VDDC	ADC1_IN3	ADC1_IN7	ADC0_IN3	ADC0_IN7	*NC*	VDDC	PHY_EXTRES	PHY_VSSAT2	PHY_TXP	PHY_TXN
M	SPI0_SIO2	SPI0_SIO3	VSS	VDDC	ADC1_IN1	ADC1_IN5	ADC1_VSS	ADC0_IN1	ADC0_IN5	VDDC	BSCAN_TRSTn	PHY_VSSACP	PHY_VDDIOAC	PHY_VDDCAP	PHY_ATP
N	SPI0_MISO	SPI0_MOSI	*NC*	ADC1_VDDIO	VSS	VSS	VDDC	ADC0_VDDIO	*NC*	VSS	VSS	VDDIO	*NC*	VSS	VDDC
P	SPI0_CLK	MMIO11 PWM7	MMIO09 PWM5	ADC1_VREFP	ADC1_IN2	ADC1_IN6	VDDIO	ADC0_IN2	ADC0_IN6	ADC0_VSS	MMIO07 PWM3	MMIO05 PWM1	MMIO03 XM_TXOE XM_TXOE_ECLK FO_SD	MMIO01 XM_TX XM_TX_ECLK FO_TD	VSS
R		MMIO10 PWM6	MMIO08 PWM4	ADC1_IN0	ADC1_IN4	ADC1_VREFM	ADC0_VREFP	ADC0_IN0	ADC0_IN4	ADC0_VREFM	MMIO06 PWM2	MMIO04 PWM0	MMIO02 XM_ECLK FB_CLK FO_FN_EN	MMIO00 FO_RD	

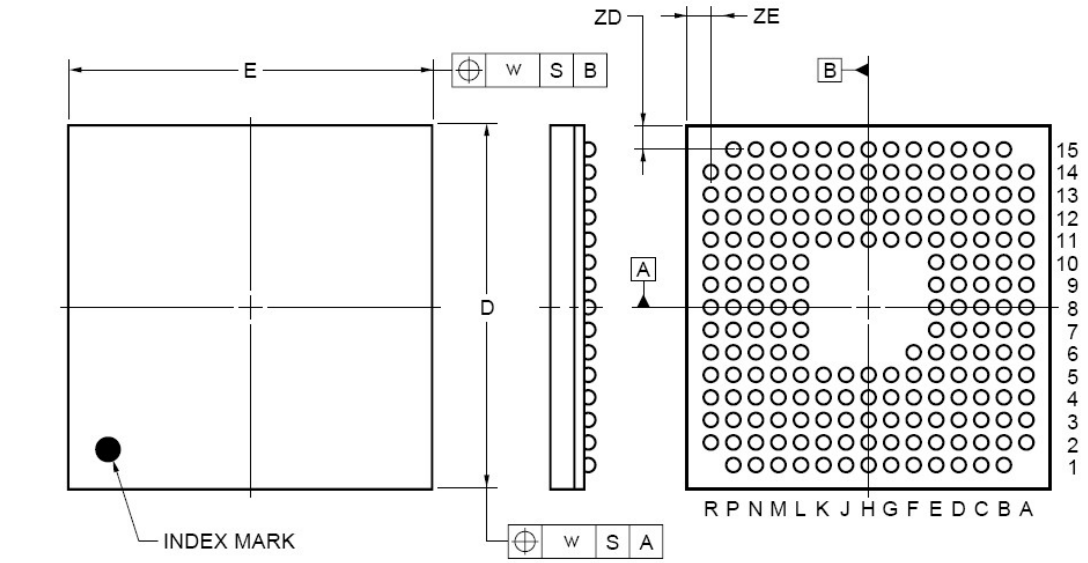
Pinning netX10 Top view

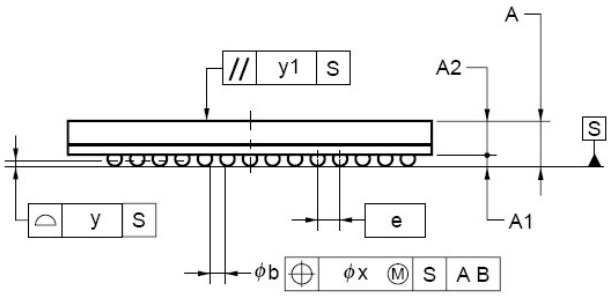
IO-Power (3.3V)
Core-Power (1.5V)
Ground
XTAL / Reset
JTAG / Test
SPI
DPM / MEM
MMIO
PHY
USB
ADC
RDY / RUN

6 Mechanical Dimensions

The netX10 comes in a 197 pin FBGA package:
(13x13 mm, 0.8mm pitch)

197-PIN PLASTIC FBGA (13x13)





ITEM	MILLIMETERS
D	13.00±0.10
E	13.00±0.10
w	0.20
A	1.48±0.10
A1	0.35±0.06
A2	1.13
e	0.80
b	$0.50^{+0.05}_{-0.10}$
x	0.08
y	0.10
y1	0.20
ZD	0.90
ZE	0.90

P197F1-80-EN3

7 Revision History

Rev	Date	Name	Revisions
0.1	15. 02. 2009	Y.Zavgorodniy	first draft
0.2	26.02.2010	J. Lipfert	Revised and updated
0.3	12.03.2010	J. Lipfert	Corrected Device Pinning for C14, D14, E14, E15, F14, F15, G14, G15, H14, H15. Corrected SQI Data Rate (Chapter 2.8)