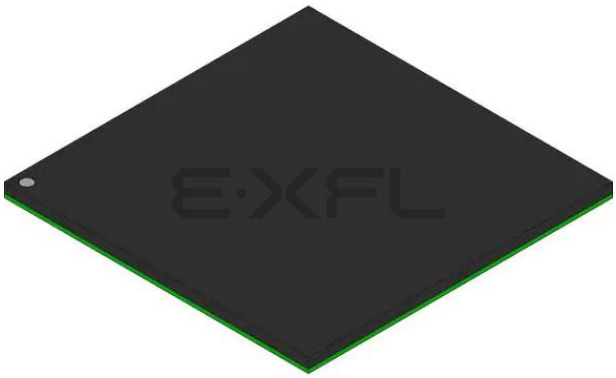

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	Coldfire V2
Core Size	32-Bit Single-Core
Speed	66MHz
Connectivity	CANbus, EBI/EMI, Ethernet, I²C, SPI, UART/USART
Peripherals	DMA, LVD, POR, PWM, WDT
Number of I/O	142
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	64K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	256-LBGA
Supplier Device Package	256-MAPBGA (17x17)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mcf5280cvf66

  - Toggle-on-overflow feature for pulse-width modulator (PWM) generation
  - One dual-mode pulse accumulation channel per timer
- Four periodic interrupt timers (PITs)
  - 16-bit counter
  - Selectable as free running or count down
- Software watchdog timer
  - 16-bit counter
  - Low-power mode support
- Phase locked loop (PLL)
  - Crystal or external oscillator reference
  - 2- to 10-MHz reference frequency for normal PLL mode
  - 33- to 80-MHz (66 MHz for MCF5214/16) oscillator reference frequency for 1:1 mode
  - Low-power modes supported
  - Separate clock output pin
- Two interrupt controllers
  - Support for up to 63 interrupt sources per interrupt controller (a total of 126), organized as follows:
    - 56 fully-programmable interrupt sources
    - 7 fixed-level interrupt sources
  - Seven external interrupt signals
  - Unique vector number for each interrupt source
  - Ability to mask any individual interrupt source or all interrupt sources (global mask-all)
  - Support for hardware and software interrupt acknowledge (IACK) cycles
  - Combinatorial path to provide wake-up from low-power modes
- DMA controller
  - Four fully programmable channels
  - Dual-address transfer support with 8-, 16- and 32-bit data capability along with support for 16-byte (4 x 32-bit) burst transfers
  - Source/destination address pointers that can increment or remain constant
  - 24-bit byte transfer counter per channel
  - Auto-alignment transfers supported for efficient block movement
  - Bursting and cycle steal support
  - Software-programmable connections between the 11 DMA requesters in the UARTs (3), 32-bit timers (4) plus external logic (4) and the four DMA channels
- External bus interface
  - Glueless connections to external memory devices (e.g., SRAM, Flash, ROM, etc.)
  - SDRAM controller supports 8-, 16-, and 32-bit wide memory devices
  - Glueless interface to SRAM devices with or without byte strobe inputs
  - Programmable wait state generator
  - 32-bit bidirectional data bus
  - 24-bit address bus
  - Up to seven chip selects available
  - Byte/write enables (byte strobes)

base address register (RAMBAR), and system control registers that include low-power and core watchdog timer control.

1.1.3 External Interface Module (EIM)

The external interface module handles the transfer of information between the internal core and memory, peripherals, or other processing elements in the external address space.

Programmable chip-select outputs provide signals to enable external memory and peripheral circuits, providing all handshaking and timing signals for automatic wait-state insertion and data bus sizing.

Base memory address and block size are programmable, with some restrictions. For example, the starting address must be on a boundary that is a multiple of the block size. Each chip select can be configured to provide read and write enable signals suitable for use with most popular static RAMs and peripherals. Data bus width (8-bit, 16-bit, or 32-bit) is programmable on all chip selects, and further decoding is available for protection from user mode access or read-only access.

1.1.4 Chip Select

Programmable chip select outputs provide a glueless connection to external memory and peripheral circuits, providing all handshaking and timing signals for automatic wait-state insertion and data bus sizing.

1.1.5 Power Management

The MCF5282 incorporates several low-power modes of operation which are entered under program control and exited by several external trigger events. An integrated Power-On Reset (POR) circuit monitors the input supply and forces an MCU reset as the supply voltage rises. The Low Voltage Detect (LVD) section monitors the supply voltage and is configurable to force a reset or interrupt condition if it falls below the LVD trip point. The RAM standby switch provides power to RAM when the supply voltage is higher than the standby voltage. If the supply voltage to chip falls below the standby battery voltage, the RAM is switched over to the standby supply.

1.1.6 General Input/Output Ports

All of the pins associated with the external bus interface may be used for several different functions. Their primary function is to provide an external memory interface to access off-chip resources. When not used for this function, all of the pins may be used as general-purpose digital I/O pins. In some cases, the pin function is set by the operating mode, and the alternate pin functions are not supported.

The digital I/O pins on the MCF5282 are grouped into 8-bit ports. Some ports do not use all eight bits. Each port has registers that configure, monitor, and control the port pins.

1.1.7 Interrupt Controllers (INTC0/INTC1)

There are two interrupt controllers on the MCF5282, each of which can support up to 63 interrupt sources for a total of 126. Each interrupt controller is organized as 7 levels with 9 interrupt sources per level. Each interrupt source has a unique interrupt vector, and 56 of the 63 sources of a given controller provide a programmable level [1-7] and priority within the level.

Table 2-1. ColdFire Core Programming Model (continued)

BDM[1]	Register	Width (bits)	Access	Reset Value	Written with MOVEC	Section/Page
0x80F	Program Counter (PC)	32	R/W	Contents of location 0x0000_0004	No	2.2.5/2-7
Supervisor Access Only Registers						
0x002	Cache Control Register (CACR)	32	R/W	0x0000_0000	Yes	2.2.6/2-7
0x004–5	Access Control Register 0–1 (ACR0–1)	32	R/W	See Section	Yes	2.2.7/2-7
0x800	User/Supervisor A7 Stack Pointer (OTHER_A7)	32	R/W	Contents of location 0x0000_0000	No	2.2.3/2-5
0x801	Vector Base Register (VBR)	32	R/W	0x0000_0000	Yes	2.2.8/2-7
0x80E	Status Register (SR)	16	R/W	0x27--	No	2.2.9/2-8
0xC04	Flash Base Address Register (FLASHBAR)	32	R/W	0x0000_0000	Yes	2.2.10/2-8
0xC05	RAM Base Address Register (RAMBAR)	32	R/W	See Section	Yes	2.2.10/2-8

1 The values listed in this column represent the Rc field used when accessing the core registers via the BDM port. For more information see Chapter 30, “Debug Support”.

2.2.1 Data Registers (D0–D7)

D0–D7 data registers are for bit (1-bit), byte (8-bit), word (16-bit) and longword (32-bit) operations; they can also be used as index registers.

NOTE

Registers D0 and D1 contain hardware configuration details after reset. See Section 2.3.4.15, “Reset Exception” for more details.

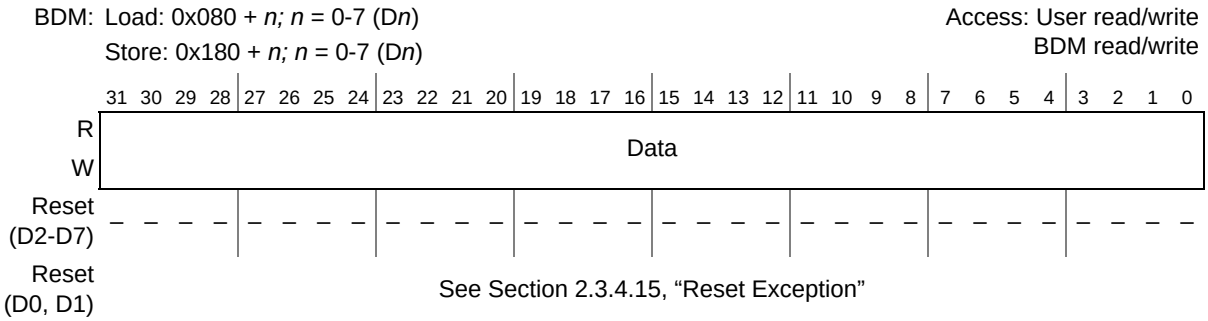

Figure 2-2. Data Registers (D0–D7)

2.2.2 Address Registers (A0–A6)

These registers can be used as software stack pointers, index registers, or base address registers. They can also be used for word and longword operations.

NOTE

Flash accesses (reads/writes) by a bus master other than the core, (DMA controller or Fast Ethernet Controller), or writes to Flash by the core during programming must use the backdoor Flash address of IPSBAR plus an offset of 0x0400_0000. For example, for a DMA transfer from the first location of Flash when IPSBAR is still at its default location of 0x4000_0000, the source register would be loaded with 0x4400_0000. Backdoor access to Flash for reads can be made by the bus master, but it takes 2 cycles longer than a direct read of the Flash if using its FLASHBAR address.

NOTE

The Flash is marked as valid on reset based on the RCON (reset configuration) pin state. Flash space is valid on reset when booting in single chip mode (RCON pin asserted and D[26]/D[17]/D[16] set to 110), or when booting internally in master mode (RCON asserted and D[26]/D[17]/D[16] are set to 111 and D[18] and D[19] are set to 00). See Chapter 27, “Chip Configuration Module (CCM)” for more details. When the default reset configuration is not overriden, the device (by default) boots in single chip mode and the Flash space will be marked as valid at address 0x0. The Flash configuration field is checked during the reset sequence to see if the Flash is secured. If it is the part will always boot from internal Flash, since it will be marked as valid, regardless of what is done for chip configuration.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	16
Field	BA31	BA30	BA29	BA28	BA27	BA26	BA25	BA24	BA23	BA22	BA21	BA20	BA19	—	
Reset	0000_0000_0000_0000														
R/W	R/W														

	15	9	8	7	6	5	4	3	2	1	0
Field	—		WP	—		C/I	SC	SD	UC	UD	V
Reset	0000_0001_0010_000										See Note
R/W	R					R/W					
Address	CPU + 0xC04										

Note: The reset value for the valid bit is determined by the chip mode selected at reset (see Chapter 27, “Chip Configuration Module (CCM)”).

Figure 6-3. Flash Base Address Register (FLASHBAR)

Table 10-4. IPRH*n* Field Descriptions

Bits	Name	Description
31–0	INT	Interrupt pending. Each bit corresponds to an interrupt source. The corresponding IMRH*n* bit determines whether an interrupt condition can generate an interrupt. At every system clock, the IPRH*n* samples the signal generated by the interrupting source. The corresponding IPRH*n* bit reflects the state of the interrupt signal even if the corresponding IMRH*n* bit is set. 0 The corresponding interrupt source does not have an interrupt pending 1 The corresponding interrupt source has an interrupt pending

	31	16
Field	INT[31:16]	
Reset	0000_0000_0000_0000	
R/W	R	

	15	1	0
Field	INT[16:1]		—
Reset	0000_0000_0000_0000		
R/W	R		
	IPSBAR + 0xC04, 0xD04		

Figure 10-2. Interrupt Pending Register Low (IPRL*n*)

Table 10-5. IPRL*n* Field Descriptions

Bits	Name	Description
31–1	INT	Interrupt Pending. Each bit corresponds to an interrupt source. The corresponding IMRL*n* bit determines whether an interrupt condition can generate an interrupt. At every system clock, the IPRL*n* samples the signal generated by the interrupting source. The corresponding IPRL*n* bit reflects the state of the interrupt signal even if the corresponding IMRL*n* bit is set. 0 The corresponding interrupt source does not have an interrupt pending 1 The corresponding interrupt source has an interrupt pending
0	—	Reserved, should be cleared.

10.3.2 Interrupt Mask Register (IMRH*n*, IMRL*n*)

The IMRH*n* and IMRL*n* registers are each 32 bits in size and provide a bit map for each interrupt to allow the request to be disabled (1 = disable the request, 0 = enable the request). The IMR*n* is set to all ones by reset, disabling all interrupt requests. The IMR*n* can be read and written. A write that sets bit 0 of the IMR forces the other 63 bits to be set, disabling all interrupt sources, and providing a global mask-all capability.

14.2.5 External Interrupt Signals

14.2.5.1 External Interrupts ($\overline{\text{IRQ}}$[7:1])

These inputs are the external interrupt sources. See Chapter 11, “Edge Port Module (EPORT)” for more information on these interrupt sources and their corresponding registers.

These pins are configured as GPIO PNQ[7:1] in single-chip mode.

14.2.6 Ethernet Module Signals

The following signals are used by the Ethernet module for data and clock signals.

NOTE

These signals are not available on the MCF5214 and MCF5216.

14.2.6.1 Management Data (EMDIO)

The bidirectional EMDIO signal transfers control information between the external PHY and the media-access controller. Data is synchronous to EMDC and applies to MII mode operation. This signal is an input after reset. When the FEC is operated in 10 Mbps 7-wire interface mode, this signal should be connected to VSS.

This pin can also be configured as GPIO PAS5 or URXD2.

14.2.6.2 Management Data Clock (EMDC)

EMDC is an output clock which provides a timing reference to the PHY for data transfers on the EMDIO signal and applies to MII mode operation.

This pin can also be configured as GPIO PAS4 or UTXD2.

14.2.6.3 Transmit Clock (ETXCLK)

This is an input clock which provides a timing reference for ETXEN, ETXD[3:0] and ETXER.

This pin can also be configured as GPIO PEH7.

14.2.6.4 Transmit Enable (ETXEN)

The transmit enable (ETXEN) output indicates when valid nibbles are present on the MII. This signal is asserted with the first nibble of a preamble and is negated before the first ETXCLK following the final nibble of the frame.

This pin can also be configured as GPIO PEH6.

14.2.6.5 Transmit Data 0 (ETXD0)

ETXD0 is the serial output Ethernet data and is only valid during the assertion of ETXEN. This signal is used for 10 Mbps Ethernet data. This signal is also used for MII mode data in conjunction with ETXD[3:1].

This pin can also be configured as GPIO PEH5.

	15	14	13	12	11	10	9	8	0
Field	—		NAM	COC	IS	RTIM		RC	
Reset	Uninitialized								
R/W	R/W								
Addr	IPSBAR + 0x040								

Figure 15-2. DRAM Control Register (DCR)

Table 15-4 describes DCR fields.

Table 15-4. DCR Field Descriptions

Bits	Name	Description
15-14	—	Reserved, should be cleared.
13	NAM	No address multiplexing. Some implementations require external multiplexing. For example, when linear addressing is required, the SDRAM should not multiplex addresses on SDRAM accesses. 0 The SDRAM controller multiplexes the external address bus to provide column addresses. 1 The SDRAM controller does not multiplex the external address bus to provide column addresses.
12	COC	Command on SDRAM clock enable (SCKE). Implementations that use external multiplexing (NAM = 1) must support command information to be multiplexed onto the SDRAM address bus. 0 SCKE functions as a clock enable; self-refresh is initiated by the SDRAM controller through DCR[IS]. 1 SCKE drives command information. Because SCKE is not a clock enable, self-refresh cannot be used (setting DCR[IS]). Thus, external logic must be used if this functionality is desired. External multiplexing is also responsible for putting the command information on the proper address bit.
11	IS	Initiate self-refresh command. 0 Take no action or issue a SELFX command to exit self refresh. 1 If DCR[COC] = 0, the SDRAM controller sends a SELF command to both SDRAM blocks to put them in low-power, self-refresh state where they remain until IS is cleared. When IS is cleared, the controller sends a SELFX command for the SDRAMs to exit self-refresh. The refresh counter is suspended while the SDRAMs are in self-refresh; the SDRAM controls the refresh period.
10–9	RTIM	Refresh timing. Determines the timing operation of auto-refresh in the SDRAM controller. Specifically, it determines the number of bus clocks inserted between a REF command and the next possible ACTV command. This same timing is used for both memory blocks controlled by the SDRAM controller. This corresponds to t_{RC} in the SDRAM specifications. 00 3 clocks 01 6 clocks 1x 9 clocks
8–0	RC	Refresh count. Controls refresh frequency. The number of bus clocks between refresh cycles is (RC + 1) x 16. Refresh can range from 16–8192 bus clocks to accommodate both standard and low-power SDRAMs with bus clock operation from less than 2 MHz to greater than 50 MHz. The following example calculates RC for an auto-refresh period for 4096 rows to receive 64 ms of refresh every 15.625 µs for each row (1031 bus clocks at 66 MHz). This operation is the same as in asynchronous mode. # of bus clocks = 1031 = (RC field + 1) x 16 RC = (1031 bus clocks/16) -1 = 63.44, which rounds to 63; therefore, RC = 0x3F.

Table 16-1. DMAREQC Field Description (continued)

31–16	—	Reserved. Should be cleared.
15–0	DMAC*n*	DMA Channel *n*. Each four bit field defines the logical connection between the DMA requestors and that DMA channel. There are seven possible requesters (4 DMA Timers and 3 UARTs). Any request can be routed to any of the DMA channels. Effectively, the DMAREQC provides a software-controlled routing matrix of the 7 DMA request signals to the 4 channels of the DMA module. DMAC3 controls DMA channel 3. DMAC2 controls DMA channel 2. DMAC1 controls DMA channel 1. DMAC0 controls DMA channel 0. 1000 UART0. 1001 UART1. 1010 UART2. 0100 DMA Timer 0. 0101 DMA Timer 1. 0110 DMA Timer 2. 0111 DMA Timer 3. All other values are reserved and will not generate a DMA request.

Table 16-3. DCR*n* Field Descriptions (continued)

Bits	Name	Description
15	AT	AT is available only if MPARK[BCR24BIT] = 1. DMA acknowledge type. Controls whether acknowledge information is provided for the entire transfer or only the final transfer. 0 Entire transfer. DMA acknowledge information is displayed anytime the channel is selected as the result of an external request. 1 Final transfer (when BCR reaches zero). For dual-address transfer, the acknowledge information is displayed for both the read and write cycles.
14–0	—	Reserved, should be cleared.

16.4.5 DMA Status Registers (DSR0–DSR3)

In response to an event, the DMA controller writes to the appropriate DSR*n* bit, Figure 16-9. Only a write to DSR*n*[DONE] results in action.

	7	6	5	4	3	2	1	0
Field	—	CE	BES	BED	—	REQ	BSY	DONE
Reset	0000_0000							
R/W	R/W							
Address	IPSBAR + 0x110, 0x150, 0x190, 0x1D0							

Figure 16-9. DMA Status Registers (DSR*n*)

Table 16-4 describes DSR*n* fields.

Table 16-4. DSR*n* Field Descriptions

Bits	Name	Description
7	—	Reserved, should be cleared.
6	CE	Configuration error. Occurs when BCR, SAR, or DAR does not match the requested transfer size, or if BCR = 0 when the DMA receives a start condition. CE is cleared at hardware reset or by writing a 1 to DSR[DONE]. 0 No configuration error exists. 1 A configuration error has occurred.
5	BES	Bus error on source 0 No bus error occurred. 1 The DMA channel terminated with a bus error during the read portion of a transfer.
4	BED	Bus error on destination 0 No bus error occurred. 1 The DMA channel terminated with a bus error during the write portion of a transfer.
3	—	Reserved, should be cleared.
2	REQ	Request 0 No request is pending or the channel is currently active. Cleared when the channel is selected. 1 The DMA channel has a transfer remaining and the channel is not selected.

IPSBAR Offset: 0x1180 Access: User read/write

	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2	1	0
R	R_DES_START								0	0
W										
Reset	— — — —	— — — —	— — — —	— — — —	— — — —	— — — —	— — — —	— —	—	—

Figure 17-22. Ethernet Receive Descriptor Ring Start Register (ERDSR)

Table 17-26. ERDSR Field Descriptions

Field	Description
31–2 R_DES_START	Pointer to start of receive buffer descriptor queue.
1–0	Reserved, must be cleared.

17.4.23 Transmit Buffer Descriptor Ring Start Registers (ETSDR)

ETSDR provides a pointer to the start of the circular transmit buffer descriptor queue in external memory. This pointer must be 32-bit aligned; however, it is recommended it be made 128-bit aligned (evenly divisible by 16). You should write zeros to bits 1 and 0. Hardware ignores non-zero values in these two bit positions.

This register is undefined at reset and must be initialized prior to operation.

IPSBAR Offset: 0x1184 Access: User read/write

	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2	1	0
R	X_DES_START								0	0
W										
Reset	— — — —	— — — —	— — — —	— — — —	— — — —	— — — —	— — — —	— —	—	—

Figure 17-23. Transmit Buffer Descriptor Ring Start Register (ETDSR)

Table 17-27. ETDSR Field Descriptions

Field	Description
31–2 X_DES_START	Pointer to start of transmit buffer descriptor queue.
1–0	Reserved, must be cleared.

17.4.24 Receive Buffer Size Register (EMRBR)

The EMRBR is a user-programmable register that dictates the maximum size of all receive buffers. This value should take into consideration that the receive CRC is always written into the last receive buffer. To allow one maximum size frame per buffer, EMRBR must be set to RCR[MAX_FL] or larger. To properly align the buffer, EMRBR must be evenly divisible by 16. To ensure this, bits 3–0 are forced low.

group address is determined by the I/G bit in the destination address field. A flowchart for address recognition on received frames appears in the figures below.

Address recognition is accomplished through the use of the receive block and microcode running on the microcontroller. The flowchart shown in Figure 17-27 illustrates the address recognition decisions made by the receive block, while Figure 17-28 illustrates the decisions made by the microcontroller.

If the DA is a broadcast address and broadcast reject (RCR[BC_REJ]) is cleared, then the frame is accepted unconditionally, as shown in Figure 17-27. Otherwise, if the DA is not a broadcast address, then the microcontroller runs the address recognition subroutine, as shown in Figure 17-28.

If the DA is a group (multicast) address and flow control is disabled, then the microcontroller performs a group hash table lookup using the 64-entry hash table programmed in GAUR and GALR. If a hash match occurs, the receiver accepts the frame.

If flow control is enabled, the microcontroller does an exact address match check between the DA and the designated PAUSE DA (01:80:C2:00:00:01). If the receive block determines the received frame is a valid PAUSE frame, the frame is rejected. The receiver detects a PAUSE frame with the DA field set to the designated PAUSE DA or the unicast physical address.

If the DA is the individual (unicast) address, the microcontroller performs an individual exact match comparison between the DA and 48-bit physical address that you program in the PALR and PAUR registers. If an exact match occurs, the frame is accepted; otherwise, the microcontroller does an individual hash table lookup using the 64-entry hash table programmed in registers, IAUR and IALR. In the case of an individual hash match, the frame is accepted. Again, the receiver accepts or rejects the frame based on PAUSE frame detection, shown in Figure 17-27.

If neither a hash match (group or individual) nor an exact match (group or individual) occur, and if promiscuous mode is enabled (RCR[PROM] set), the frame is accepted and the MISS bit in the receive buffer descriptor is set; otherwise, the frame is rejected.

Similarly, if the DA is a broadcast address, broadcast reject (RCR[BC_REJ]) is asserted, and promiscuous mode is enabled, the frame is accepted and the MISS bit in the receive buffer descriptor is set; otherwise, the frame is rejected.

In general, when a frame is rejected, it is flushed from the FIFO.

- Start/end break interrupt/status

23.2 External Signal Description

Table 23-1 briefly describes the UART module signals.

Table 23-1. UART Module External Signals

Signal	Description
UTXD*n*	Transmitter Serial Data Output. UTXD*n* is held high (mark condition) when the transmitter is disabled, idle, or operating in the local loopback mode. Data is shifted out on UTXD*n* on the falling edge of the clock source, with the least significant bit (lsb) sent first.
URXD*n*	Receiver Serial Data Input. Data received on URXD*n* is sampled on the rising edge of the clock source, with the lsb received first.
$\overline{\text{UCTS}n}$	Clear-to- Send. This input can generate an interrupt on a change of state.
$\overline{\text{URTS}n}$	Request-to-Send. This output can be programmed to be negated or asserted automatically by the receiver or the transmitter. When connected to a transmitter's $\overline{\text{UCTS}n}$, $\overline{\text{URTS}n}$ can control serial data flow.

Figure 23-2 shows a signal configuration for a UART/RS-232 interface.

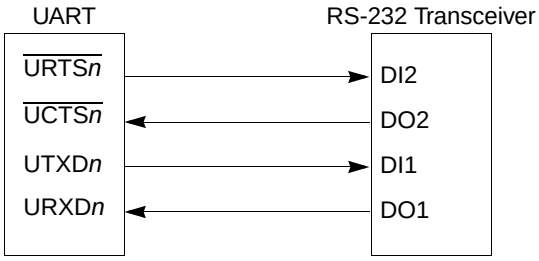

Figure 23-2. UART/RS-232 Interface

23.3 Memory Map/Register Definition

This section contains a detailed description of each register and its specific function. Flowcharts in Section 23.5, "Initialization/Application Information," describe basic UART module programming. Writing control bytes into the appropriate registers controls the operation of the UART module.

NOTE

UART registers are accessible only as bytes.

NOTE

Interrupt can mean an interrupt request asserted to the CPU or a DMA request.

Table 26-1. Ports External Signals

Primary Function (Pin Name)[1]	GPIO (Default Function)	Alternate Function 1	Alternate Function 2	Description
D[31:0][2]	PA, PB, PC, PD	—	—	External data bus / Ports A, B, C, D
$\overline{OE}$[1]	PE[7]	—	—	Output enable for external reads / Port E[7]
$\overline{TA}$[1]	PE[6]	—	—	Transfer acknowledge for external data transfer / Port E[6]
$\overline{TEA}$[1]	PE[5]	—	—	Transfer error acknowledge for external data transfer / Port E[5]
R/$\overline{W}$[1]	PE[4]	—	—	Read/Write indication for external data transfer / Port E[4]
SIZ1[1]	PE[3]	SYNCA	—	Size of the external data transfer / Port E[3] / Timer A sync
SIZ0[1]	PE[2]	SYNCB	—	Size of the external data transfer / Port E[3:2] / Timer B sync
$\overline{TS}$[1]	PE[1]	SYNCA	—	Transfer start indication for external data transfer / Port E[1] / Timer A sync
$\overline{TIP}$[1]	PE[0]	SYNCB	—	Transfer in progress indication for external data transfer / Port E[0] / Timer B sync
A[23:21][1]	PF[7:5]	$\overline{CS}$[6:4]	—	External address bus [23:21] / Port F[7:5] / Chip selects 6–4
A[20:0][1]	PF[4:0], PG, PH	—	—	External address bus [20:0] / Ports F[4:0], G, H
$\overline{BS}$[3:0][1]	PJ[7:4]	—	—	Byte strobes for external data transfer / Port J[7:4] / SDRAM column address strobes
$\overline{CS}$[3:0][1]	PJ[3:0]	—	—	Chip selects 3 - 0 / Port J[3:0]
DDATA[3:0]	PDD[7:4]	—	—	Debug data / Port DD[7:4]
PST[3:0]	PDD[7:4]	—	—	Processor status / Port DD[3:0]
CANRX	PAS[3]	URXD2	—	FlexCAN receive data / Port AS[3] / URXD2
CANTX	PAS[2]	UTXD2	—	FlexCAN transmit data / Port AS[2] / UTXD2
SDA	PAS[1]	URXD2	—	I^2C serial data / Port AS[1] / URXD2
SCL	PAS[0]	UTXD2	—	I^2C serial clock / Port AS[0] / UTXD2
IRQ[7:1][3]	PNQ[7:1]	—	—	Edge Port external interrupt pins / Port NQ[7:1]
AN[56:55:53:52][2]	PQA [4:3:1:0]	ETRIG[2:1], MA[1:0]	—	QADC analog inputs / Port QA[4:3:1:0] / external triggers / external multiplex control
AN[3:0] [2]	PQB[3:0]	ANZ, ANY, ANX, ANW	—	QADC analog inputs / Port QB[3:0] / multiplexed analog inputs
QSPI_CS [3:0]	PQS[6:3]	—	—	QSPI synchronous peripheral chip selects / Port QS[3:6]
QSPI_CLK	PQS[2]	—	—	QSPI serial clock / Port QS[2]
QSPI_DIN	PQS[1]	—	—	QSPI serial data input / Port QS[1]
QSPI_DOUT	PQS[0]	—	—	QSPI serial data output / Port QS[0]
$\overline{SRAS}$	PSD[5]	—	—	SDRAM synchronous row address strobe / Port SD[5]

26.3.2 Register Descriptions

26.3.2.1 Port Output Data Registers (PORT*n*)

The PORT*n* registers store the data to be driven on the corresponding port *n* pins when the pins are configured for digital output.

Most PORT*n* registers have a full 8-bit implementation, as shown in Figure 26-3. The remaining PORT*n* registers use fewer than eight bits. Their bit definitions are shown in Figure 26-4, Figure 26-5, and Figure 26-6.

At reset, all bits in the PORT*n* registers are set.

Reading a PORT*n* register returns the current values in the register, not the port *n* pin values.

PORT*n* bits can be set by setting the PORT*n* register, or by setting the corresponding bits in the PORT*n*P/SET*n* register. They can be cleared by clearing the PORT*n* register, or by clearing the corresponding bits in the CLR*n* register.

	7	6	5	4	3	2	1	0
Field	PORT*n*7	PORT*n*6	PORT*n*5	PORT*n*4	PORT*n*3	PORT*n*2	PORT*n*1	PORT*n*0
Reset	1111_1111							
R/W:	R/W							
Address	IPSBAR + 0x10_0000 (PORTA), 0x10_0001 (PORTB), 0x10_0002 (PORTC), 0x10_0003 (PORTD), 0x10_0004 (PORTE), 0x10_0005 (PORTF), 0x10_0006 (PORTG), 0x10_0007 (PORTH), 0x10_0008 (PORTJ), 0x10_0009 (PORTDD), 0x10_000A (PORTEH), 0x10_000B (PORTEL)							

Figure 26-3. Port Output Data Registers (8-bit)

	7	6	5	4	3	2	1	0
Field	—	PORT*n*6	PORT*n*5	PORT*n*4	PORT*n*3	PORT*n*2	PORT*n*1	PORT*n*0
Reset	0011_1111							
R/W:	R	R/W						
Address	IPSBAR + 0x10_000D (PORTQS)							

Figure 26-4. Port Output Data Register (7-bit)

	7	6	5	4	3	2	1	0
Field	—		PORT*n*5	PORT*n*4	PORT*n*3	PORT*n*2	PORT*n*1	PORT*n*0
Reset	0011_1111							
R/W:	R	R/W						
Address	IPSBAR + 0x10_000C (PORTAS), 0x10_000E (PORTSD)							

Figure 26-5. Port Output Data Registers (6-bit)

Table 27-8. Configuration During Reset[1] (continued)

Pin(s) Affected	Default Configuration	Override Pins in Reset[2,3,4]	Function
CS0	RCON[4:3] = 00 RCON2 = 0	**D[19:18]**	**Boot Device**
		00	Internal with 32-bit port[5]
		10	External with 8-bit port
		01	External with 16-bit port
		11	External with 32-bit port
All output pins	RCON5 = 1	**D21**	**Output Pad Drive Strength**
		0	Partial strength
		1	Full strength[5]
Clock mode	N/A	**CLKMOD1, CLKMOD0**	**Clock Mode**
		00	External clock mode (PLL disabled)
		01	1:1 PLL mode
		10	Normal PLL mode with external clock reference
		11	Normal PLL mode w/crystal reference
A[23:21]/$\overline{CS}$[6:4]	RCON[9:8] = 00	**D[25:24]**	**Chip Select Configuration**
		00	PF[7:5] = A[23:21][5]
		10	PF[7] = $\overline{CS6}$ / PF[6:5] = A[22:21]
		01	PF[7:6] = $\overline{CS6}$, $\overline{CS5}$ / PF[5] = A[21]
		11	PF[7:6] = $\overline{CS6}$, $\overline{CS5}$, $\overline{CS4}$

1 Modifying the default configurations is possible only if the external $\overline{RCON}$ pin is asserted.

2 The D[31:27, 23:22, 20, 15:0] pins do not affect reset configuration.

3 The external reset override circuitry drives the data bus pins with the override values while $\overline{RSTO}$ is asserted. It must stop driving the data bus pins within one CLKOUT cycle after $\overline{RSTO}$ is negated. To prevent contention with the external reset override circuitry, the reset override pins are forced to inputs during reset and do not become outputs until at least one CLKOUT cycle after $\overline{RSTO}$ is negated.

4 RCON[0] has higher priority than RCON[3:2]. When RCON[0] is configured to boot the chip in single chip mode, the part will boot internally with a 32-bit port overriding any configuration set by RCON[3:2].

5 Default configuration

27.6.2 Chip Mode Selection

The chip mode is selected during reset and reflected in the MODE field of the chip configuration register (CCR). See Section 27.5.3.1, “Chip Configuration Register (CCR).” Once reset is exited, the operating mode cannot be changed. Table 27-9 shows the mode selection during reset configuration.

[1] All channels not listed are reserved or unimplemented and return undefined results.

28.6.8 Result Registers

The result word table is a 64 half-word (128 byte) long by 10-bit wide RAM. An entry is written by the QADC after completing an analog conversion specified by the corresponding CCW table entry.

28.6.8.1 Right-Justified Unsigned Result Register (RJURR)

	15 … 10	9 … 8
Field	—	RESULT
Reset	0000_00	Undefined
R/W:	R	R/W

	7 … 0
Field	RESULT
Reset	Undefined
R/W:	R/W
Address	IPSBAR + 0x19_0280, 0x19_02fe

Figure 28-15. Right-Justified Unsigned Result Register (RJURR)

Table 28-18. RJURR Field Descriptions

Bit(s)	Name	Description
15–10	—	Reserved, should be cleared.
9–0	RESULT	The conversion result is unsigned, right-justified data.

28.6.8.2 Left-Justified Signed Result Register (LJSRR)

	15	14 … 8
Field	S	RESULT
Reset	Undefined	
R/W:	R/W	

	7 … 0
Field	RESULT
Reset	Undefined
R/W:	R/W
Address	IPSBAR + 0x19_0300, 0x19_037e

Figure 28-16. Left-Justified Signed Result Register (LJSRR)

28.9.4 Analog Supply Filtering and Grounding

Two important factors influencing performance in analog integrated circuits are supply filtering and grounding. Generally, digital circuits use bypass capacitors on every V_{DD}/V_{SS} signal pair. This applies to analog subsystems and submodules also. Equally important as bypassing is the distribution of power and ground.

Analog supplies should be isolated from digital supplies as much as possible. This necessity stems from the higher performance requirements often associated with analog circuits. Therefore, deriving an analog supply from a local digital supply is not recommended. However, if for cost reasons digital and analog power are derived from a common regulator, filtering of the analog power is recommended in addition to the bypassing of the supplies already mentioned. For example, an RC low pass filter could be used to isolate the digital and analog supplies when generated by a common regulator. If multiple high precision analog circuits are locally employed (for example, two A/D converters), the analog supplies should be isolated from each other as sharing supplies introduces the potential for interference between analog circuits.

Grounding is the most important factor influencing analog circuit performance in mixed signal systems (or in standalone analog systems). Close attention must be paid not to introduce additional sources of noise into the analog circuitry. Common sources of noise include ground loops, inductive coupling, and combining digital and analog grounds together inappropriately.

The problem of how and when to combine digital and analog grounds arises from the large transients which the digital ground must handle. If the digital ground is not able to handle the large transients, the associated current can return to ground through the analog ground. It is this excess current overflowing into the analog ground which causes performance degradation by developing a differential voltage between the true analog ground and the microcontroller’s ground pins. The end result is that the ground observed by the analog circuit is no longer true ground and thus skews converter performance.

Two similar approaches to improving or eliminating the problems associated with grounding excess transient currents involve star-point ground systems. One approach is to star-point the different grounds at the power supply origin, thus keeping the ground isolated. Refer to Figure 28-49.

Another approach is to star-point the different grounds near the analog ground signal on the microcontroller by using small traces for connecting the non-analog grounds to the analog ground. The small traces are meant only to accommodate dc differences, not ac transients.

NOTE

This star-point scheme still requires adequate grounding for digital and analog subsystems in addition to the star-point ground.

Table A-3. Register Memory Map (continued)

Address	Name	Mnemonic	Size
IPSBAR + 0xC18	Interrupt Level Request Register 0	ILRR0	8
IPSBAR + 0XC19	Interrupt Acknowledge Level and Priority Register 0	IACKLPR0	8
IPSBAR + 0xC41	Interrupt Control Register 0-01	ICR001	8
IPSBAR + 0xC42	Interrupt Control Register 0-02	ICR002	8
IPSBAR + 0xC43	Interrupt Control Register 0-03	ICR003	8
IPSBAR + 0xC44	Interrupt Control Register 0-04	ICR004	8
IPSBAR + 0xC45	Interrupt Control Register 0-05	ICR005	8
IPSBAR + 0xC46	Interrupt Control Register 0-06	ICR006	8
IPSBAR + 0xC47	Interrupt Control Register 0-07	ICR007	8
IPSBAR + 0xC48	Interrupt Control Register 0-08	ICR008	8
IPSBAR + 0xC49	Interrupt Control Register 0-09	ICR009	8
IPSBAR + 0xC4A	Interrupt Control Register 0-10	ICR010	8
IPSBAR + 0xC4B	Interrupt Control Register 0-11	ICR011	8
IPSBAR + 0xC4C	Interrupt Control Register 0-12	ICR012	8
IPSBAR + 0xC4D	Interrupt Control Register 0-13	ICR013	8
IPSBAR + 0xC4E	Interrupt Control Register 0-14	ICR014	8
IPSBAR + 0xC4F	Interrupt Control Register 0-15	ICR015	8
IPSBAR + 0xC51	Interrupt Control Register 0-17	ICR017	8
IPSBAR + 0xC52	Interrupt Control Register 0-18	ICR018	8
IPSBAR +0xC53	Interrupt Control Register 0-19	ICR019	8
IPSBAR + 0xC54	Interrupt Control Register 0-20	ICR020	8
IPSBAR + 0xC55	Interrupt Control Register 0-21	ICR021	8
IPSBAR + 0xC56	Interrupt Control Register 0-22	ICR022	8
IPSBAR + 0xC57	Interrupt Control Register 0-23	ICR023	8
IPSBAR + 0xC58	Interrupt Control Register 0-24	ICR024	8
IPSBAR + 0xC59	Interrupt Control Register 0-25	ICR025	8
IPSBAR + 0xC5A	Interrupt Control Register 0-26	ICR026	8
IPSBAR + 0xC5B	Interrupt Control Register 0-27	ICR027	8
IPSBAR + 0xC5C	Interrupt Control Register 0-28	ICR028	8
IPSBAR + 0xC5D	Interrupt Control Register 0-29	ICR029	8
IPSBAR + 0xC5E	Interrupt Control Register 0-30	ICR030	8
IPSBAR + 0xC5F	Interrupt Control Register 0-31	ICR031	8
IPSBAR + 0xC60	Interrupt Control Register 0-32	ICR032	8

D

E