MTC-20280

ISDN / IDSL Terminal Controller

Data Sheet

Preliminary Rev 2.0 November 1998

Key Features

- ▼ **Integrated ARM7TDMI RISC processor core**
- ▼ **16 or 8 bit memory bus**
- ▼ **3, full-duplex HDLC formatters with FIFO**
- ▼ **3-way GCI interface and router**
- ▼ **Supports up to 8 GCI peripherals per port**
- ▼ **UART with full-duplex 16 byte FIFOs**
- ▼ **8-bit and 4-bit parallel I/O ports**
- ▼ **100 pin PQFP Package style**

Key Applications

- ▼ **ISDN NT+**
- ▼ **ISDN / IDSL routers / multiplexers**
- ▼ **ISDN PABX**
- ▼ **ISDN / IDSL Terminal Adaptors**

General Description

The MTC-20280 is a fully integrated controller for ISDN / IDSL terminal equipment applications. It has been specifically designed for control and interface functions between an ISDN access device, such as the MTC-20276 / 77 Integrated NT, and other terminal functions such as analog interfaces, e.g. by means of the MTK-40131 Short-Haul POTS chipset. It incorporates an ARM7TDMI RISC core, which can perform all of the terminal control functions required in software. It can thus form the core of an Intelligent NT, or "NTplus" unit, as well as being the core of router / multiplexer equipment for IDSL applications. the on-chip GCI router allows any B-channel in any timeslot of any GCI port to be routed to any other B-channel without the need for CPU intervention. In addition, the 3 GCI ports support all modes including the x8 multiplex mode, allowing up to 16 analog channels to be accessed per port. It can also form the core of a small ISDN based PABX.

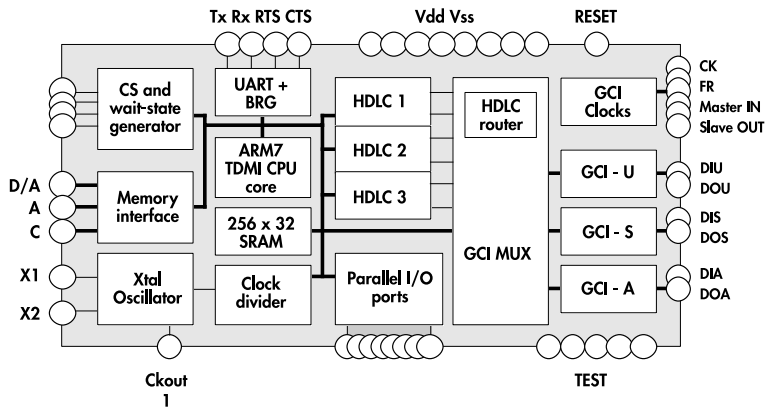


Figure 1 : Block Diagram

Ordering Information

Part number	Package	Temp.
MTC-20280PQ-C	100 pin PQFP	0 /+70°C
MTC-20280PQ-I	100 pin PQFP	- 40 /+85°C

For tape and reel, add T after package code (eg PQT)

▼

ALCATEL

TABLE OF CONTENTS

LIST OF FIGURES

Electrical Characteristics

The I/O pin type as well as the rated buffer current is given in the pin description section; note also that some inputs are Schmitt Trigger inputs.

TTL DC Electrical Characteristics (1):
(see pin description section and special notes on 5V tolerant pins)

Symbol	Parameter	Conditions	Min	Max	Unit
VIH	High Level Input Voltage		2.0		V
VIL	Low Level Input Voltage			0.8	V
VOH	High Level Output Voltage	Ioh=rated buffer current	2.4		V
VOL	Low Level Output Voltage	Iol=rated buffer current		0.4	V
Vt+	Schmitt trigger rising threshold		1.3	1.9	V
Vt-	Schmitt trigger falling threshold		0.8	1.3	V
CIN	Input Capacitance, all inputs			1	pF
COUT	Load Capacitance, all outputs			100	pF

CMOS DC Electrical Characteristics (1):
(see pin description section and special notes on 5V tolerant pins)

Symbol	Parameter	Conditions	Min	Max	Unit
VIH	High Level Input Voltage		80% of VDD		V
VIL	Low Level Input Voltage			20% of VDD	V
VOH	High Level Output Voltage	Ioh=rated buffer current	85% of VDD		V
VOL	Low Level Output Voltage	Iol=rated buffer current		0.4	V
Vt+	Schmitt trigger rising threshold		1.8	2.5	V
Vt-	Schmitt trigger falling threshold		0.8	1.3	V
CIN	Input Capacitance, all inputs			1	pF
COUT	Load Capacitance, all outputs			100	pF

(1) Rated for Vdd=2.7V to 3.6V and ambient temperature from 0 to +20 °C for C-version, -40°C to +85°C for I-version.

AC Characteristics

Memory Interface

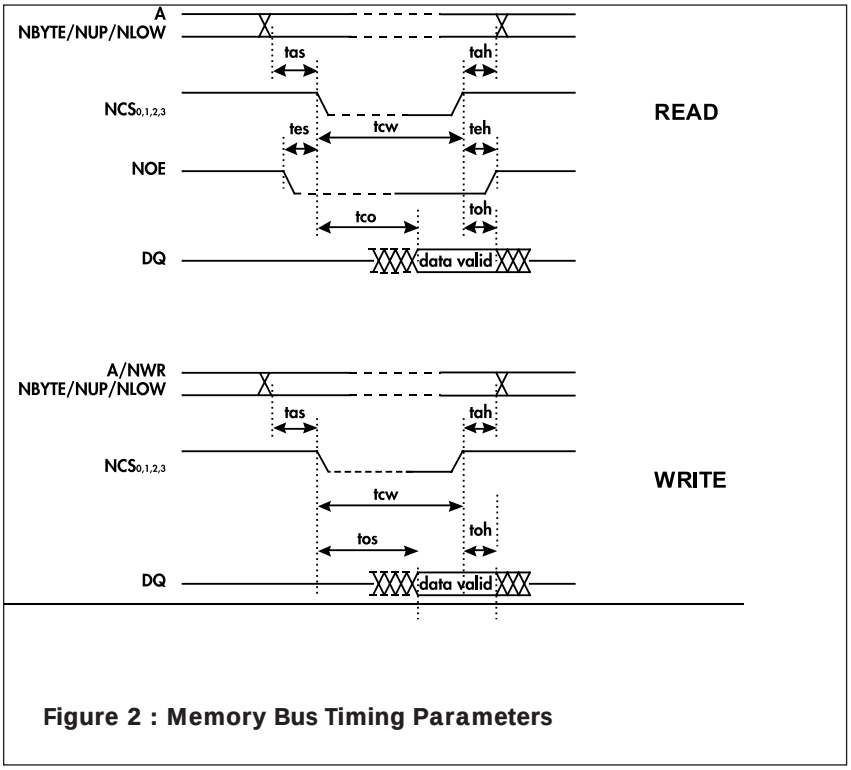


Figure 2 : Memory Bus Timing Parameters

Cycle	Parameter	Min	Max	Units
READ	tas	8		ns
	tah	0		ns
	tcw	32.5	(0.5 + x) * ASB period (x = wait cycles)	ns
	tes	0		ns
	the	0		ns
	tco		(tcw - 17)	ns
	toh	0		ns
WRITE	tas	8		ns
	tah	0		ns
	tcw	32.5	(0.5 + x) * ASB period (x = wait cycles)	ns
	tos		20	ns
	toh	0		ns

Remark: the timings are valid over all operating range conditions

GCI Interfaces

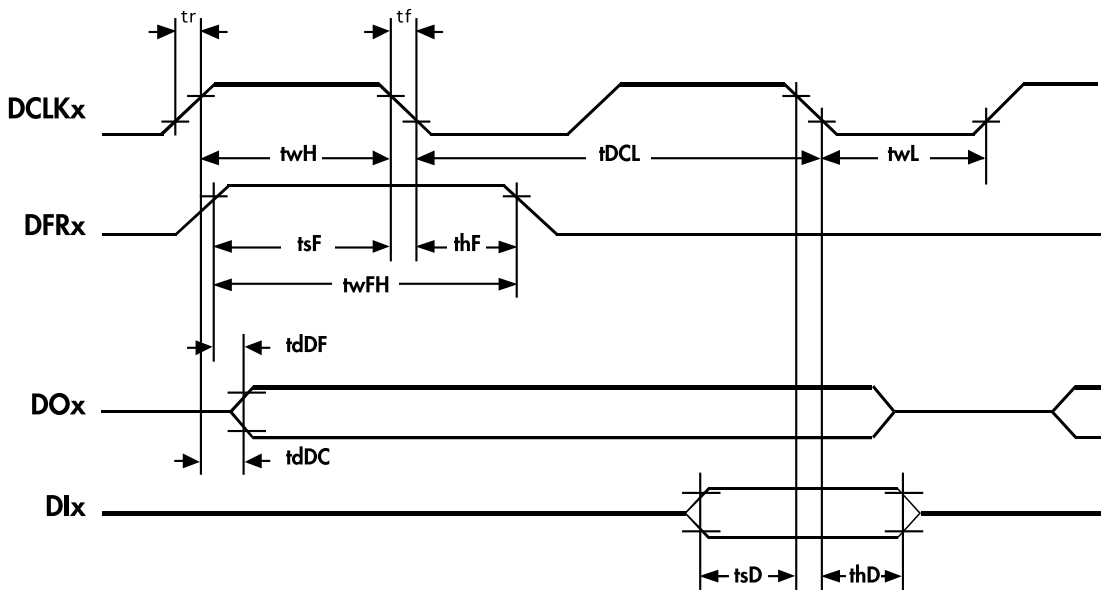


Figure 3 : GCI Bus Signal Timing

Timing reference voltages

Signal	High	Low
Output	2.4V	0.4V
Input	2.0V	0.8V

Parameter	Signal	Mnem.	Units	Min.	Max.
Clock period	DCLK	tDCL	ns	239	
Pulse width	DCLK	twL, twH	ns	90	
Frame	DFR	tsF	ns	70	tDCL-50
Frame rise / fall	DFR	tr, tf	ns		60
Frame width H	DFR	twFH	ns	130	
Frame width L	DFR	twFL	ns	tDCL	
Frame hold	DFR	tfH	ns	50	
Data delay, clock	DOUTx	tdDC	ns		100 (1)
Data delay, frame	DOUTx	tdDF	ns		150 (1)
Data set-up	DIN	tsD	ns	twH+20	
Data hold	DIN	thD	ns	50	

Note (1). Capacitive load = 150 pF.

Absolute Maximum Ratings

Stresses above those listed below can cause permanent device failure. Exposure to absolute maximum ratings for extended periods can effect device reliability. See Operating ranges section.

Symbol	Conditions	Min	Max	Unit
VDD	power supply voltage	VSS - 0.3	4.0	V
VIN	input voltage on any pin	VSS - 0.3	VDD + 0.3 AND < 4.0	V
VIN5	input voltage on any 5V tolerant pin	TBD	TBD	V

Operating Ranges

Operating ranges define the limits for functional operation and schematic characteristics of the device as described above, and for the reliability specifications as listed in the relevant section. Functionality outside these limits is not implied.

Total cumulative dwell time outside the normal power supply voltage range or the ambient temperature under bias must be less than 0.1% of the useful life as defined in the relevant section.

Furthermore, when the 5V tolerant IO cells are used in a 5V system, the application should be designed such that no 5V signals are applied when the 3.3V power supply is not present. This otherwise limits the lifetime of the device. However, if the accumulated time when this situation occurs doesn't exceed 5 hours (18000 s) over the total life of the device, the impact on the total lifetime will remain negligible. Therefore the 5V and 3.3V power supplies must always be present together, except during the short power on / off transient states which rarely occur.

Symbol	Conditions	Min	Max	Unit
VDD	power supply	3.0	3.6	V
P^{TOT} (1)	full-operation power consumption		100	mW
P^{PD} (2)	stand-by power consumption		40	mW
T^{amb}	ambient temperature I-version / C-version	-40 / 0	85 / 70	deg C

(1) Power with all blocks of the MTC-20280 operational and maximum clock frequencies used at nominal power supply voltage (3.3V) + 5%

(2) Power with all blocks of the MTC-20280 operational, except the ARM which is in power down mode, at nominal power supply voltage (3.3V) + 5%

Operating Environment

The components are intended for application in equipment for indoor operation without forced cooling airflow, only convection.

Storage and Transportation Conditions

The rated storage and transportation temperature range prior to printed board assembly is -55 to +110 °C. In the case of IC deliveries in dry bag, the conditions of time and humidity during storage are specified in Alcatel Microelectronics spec 16650. In the case of IC deliveries not in dry bag, the conditions for a maximum storage period of 2 years are as follows:

Ambient Temperature (deg C)	Relative Humidity (%)
20	80
30	70
40	60
50	50

Application Information

Typical Application

The MTC-20280 offers a flexible processor architecture for use in a wide range of telecommunications applications where the GCI standard interconnection bus is used. Particularly, it offers functions and features specifically relevant to ISDN terminal adapter and advanced NT applications such as “NTplus” or “NTN”. In such applications, the block schematic shown below is commonly used. Here, the MTC-20280 is used to control and monitor the MTC-20276 (or MTC-20277) Integrated NT device (operating in its external access mode) and the MTK-40131 chipset for short-range analog telephone (“POTS”) interface.

Other applications such as small PABX systems or remote access concentrators can also be addressed.

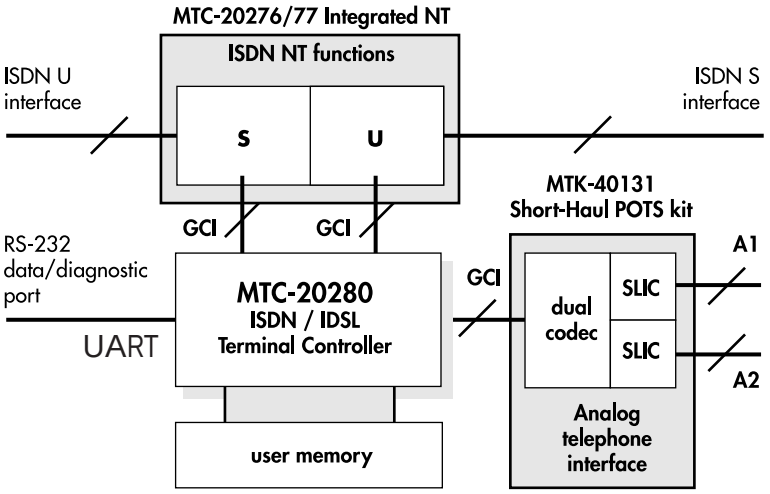


Figure 4 : Typical Application Block Diagram

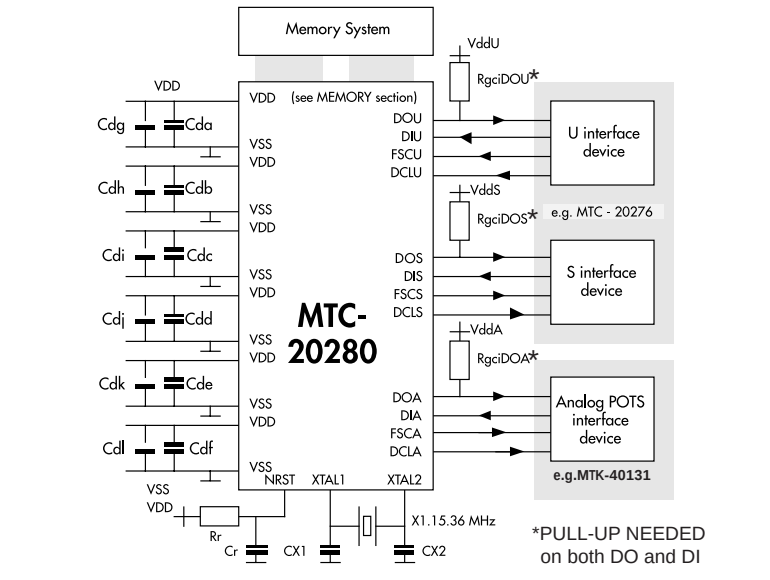


Figure 5 : Application Schematic and External Components

Application Schematic and External Components

The following external components may be needed, depending upon the application environment. GCI related components(The pull-up on the DCI DI pins) depend upon the requirements of the external GCI device:

Name	Description	Recommended Value	Tolerance
RgciDOU	Pull-up for GCI data-up output towards U interface (either to 5V or 3.3V, dependent on application)	(2kOhm)	10%
RgciDOS	Pull-up for GCI data-down output towards S interface (either to 5V or 3.3V, dependent on application)	(2kOhm)	10%
RgciDOA	Pull-up for GCI data-down output towards A interface (either to 5V or 3.3V, dependent on application)	(2Kohm)	10%

XTAL input related components:

Name	Description	Recommended Value	Tolerance
XTAL	External crystal to control on-chip oscillator via the XTAL1 and XTAL2 pins. (Alternative is to control the XTAL1 pin by an external master) clock source and connect XTAL2 to GND)	15.36 MHz	±50 ppm
CX1, CX2	Capacitors for external crystal	30 pF	±10 pF

Power supply decoupling related components:

Name	Description	Recommended Value	Tolerance
Cd[a..f]	Decoupling capacitors to be placed in between each VDD-VSS pair of MTC-20280	100 nF	10%
Cd[g..l]	Decoupling capacitors to be placed	10 µF	10%

Hardware reset related components:

Name	Description	Value	Tolerance
Rr.Cr	Power Reset circuit *	>= 1ms	10%

* Caution: As the NRST pin can be pulled LOW by a watchdog timeout, the application circuit should contain a 10 Ohm resistor in series with the pin when an external capacitor with value larger than 100nF is used. (The NRST pin will attempt to discharge this capacitor very quickly, causing a high peak current which may result in damage to the pin driver. The resistor limits the current to safe values).

The JTAG Test Port

The MTC-20280 has a standard JTAG interface to facilitate device production testing. However, it is also directly compatible with the In-Circuit Emulation hardware of ARM Ltd. It can be used to provide full de-bugging facilities, as supported by the ARM Software Development Toolkit from ARM. Please contact your Alcatel Microelectronics sales-office or representative for more information.

Detailed Functional Description

Overview

CPU
The integrated ARM7TDMI CPU will generally use the 16-bit data bus mode 'Thumb'. The external bus interface supports 16- or 32-bit transfers, multiplexed to 16- or 8 bits. Access to the on-chip SRAM can take place as 8, 16, or 32 bit transfers (it thus supports the full performance of the ARM CPU). The CPU will generally run at the clock frequency set by the crystal oscillator (15.36 Mhz). However, a programmable divider is provided to allow software control of the processor speed, and therefore the power consumption.

Clock generation and control
A master clock oscillator, based on an external crystal of 15.36Mhz, is provided. This provides an output at the crystal frequency for use by the ISDN chip (INTT or INTQ).
A programmable divider allows lower frequencies to be output, as required. The CPU clock frequency is SW selectable, allowing the power-consumption to be reduced at times when full processing speed is not required.

Memory bus
The external memory bus supports either 8-bit (for low cost) or 16 bit (high-speed) memory systems. It allows read/write access to off-chip memory and I/O resources, and includes a simple to use on-chip memory decoding scheme to minimize external logic. It is designed to interface to standard FLASH EEPROM and (pseudo)static RAMs. The bus interface logic includes a programmable WAIT STATE generator, to allow access to slow external memory or peripherals. Bus timing is to allow zero wait state execution (with fast off-chip memory) at a CPU clock of 15 Mhz. 1 kbyte of fast (0 wait-state with 32 bit access), on-chip static RAM is included.

A programmable Chip-Select ("CS") decoder defines the external memory-map - the default map ensures that the CPU can start up from reset by enabling ROM at address 0. It provides for seven external memory ranges to be individually decoded. With regard to EMC requirements, the slope of the memory bus transitions is controlled, in a manner consistent with achieving the required bus transfer speed. Each CS memory range has programmable wait-states.

An external interrupt request pin allows external peripheral devices to communicate asynchronously with the CPU.

3-way GCI interface
(Terminology. 'DOWNSTREAM' refers to the transfer of data coming from the U interface towards the S or Analog interfaces in an ISDN application. Upstream is the direction from the S or analog interfaces towards the U.)

The device provides for three GCI ports: normally allocated as follows:
1. U interface of MTC-20276/20277 INT, GCI-U
2. S interface of MTC-20276/20277 INT, GCI-S
3. Interface to analog devices such as MTK-40131 short-haul POTS chipset, GCI-A

In reality, all three GCI ports are identical - the allocation to U, S, and A (analog) is arbitrary, for clarity only.

The U interface block of the INT will always provide the GCI clocks (master) when active. (This can be achieved by issuing the AWAKE command on the GCI C/I bits to the U interface, which activates the timing generator of the U interface without actually initiating transmission).
All other GCI buses will generally be slaved to this one. In applications where the use of the U interface is not mandatory (e.g. in a micro-PABX system which allows internal calling without U activation), an internal GCI clock source can be selected. An integrated PLL system may be enabled to allow the internally generated GCI clocks to track and lock to the U GCI clock, should this become active in the course of operation.

All bytes of the GCI frames of all three GCI interfaces are accessible to the processor, for both reading and writing. A sophisticated router allows any of the GCI fields (B channel, D channel, C/I bits, and Monitor channel) to be routed to the corresponding field of any destination channel. Bytes can also be 'disabled', in which case they remain at the idle - logic '1' - state. Particularly powerful is the ability to set fixed routes of the B channels from a source to any destination without the need for further intervention by the CPU, thus relieving the CPU of much real-time processing. Up to eight GCI time-slots is supported on each GCI port independently, where external GCI clocks are available. The internal GCI clock supports one timeslot or eight timeslots. The clock source (which determines the number of timeslots supported by the channel) is independently selectable for each GCI port. Using an external clock thus allows the GCI ports to interface to all commonly used ISDN devices.

With regard to EMC requirements, the slope of the GCI data output pins is controlled, in a manner consistent with achieving the required bus transfer speed.

HDLC controllers

The three integrated HDLC controllers can be routed two / from any B or D channel of any port. In addition, they each have full-duplex 64 byte FIFOs, which allow a large timing latency and thus ease software timing constraints. The HDLC controller protocol may be disabled under software control, thus allowing the FIFOs to be used to buffer real-time data, e.g. for the processing of voice-band signals on B-channels (DTMF decoding, modem emulation, pre-recorded voice announcements etc.)

Generally, HDLC1 will be used to manage the ISDN D-chanel. D-channel conflicts between the S bus and the HDLC1 controller of the device are handled by forcing a "D-channel busy" condition on the S-bus by means of the appropriate command to the S interface of the INT. This is done only after the microprocessor has verified that the BUSY bit in the SIC's control registers is clear (i.e. D-channel not in use). HDLC controllers 2 and 3 are generally used to handle packetised data transport over the B channels (including balanced applications such as LAPB). However, in specific applications such as internal call transfer support or PABX, the D-channel to/from the S-bus requires independent management (while still monitoring the D channel to/from the U interface). HDLC 2 or 3 may be used for this purpose.

DTMF decoding

This function may be performed by low-cost, external DTMF decoder circuits, interfacing to the CPU via an on-chip parallel I/O port (programmable bit directions). Alternatively, software algorithms on the ARM7TDMI processor may be used. The 0-wait-state on-chip RAM facilitates this.

Serial I/O

A UART with selectable baud-rate and 16 byte FIFOs is provided. The baud-rate is programmable to standard rates up to 57.7kbit/s, 115kbps and 230 kbps, and is compatible with the standard UART 16C550. The Rx and Tx pins are 5V compatible. The modem controls RTS and CTS from the UART block are available as 5V compatible pins.

Parallel I/O ports

A number of 4-bit parallel I/O ports are provided, primarily to allow an interface to external DTMF decoder chips. The ports are addressable by the CPU as a latched output, an unlatched input, and a data-direction register which is used to select the direction (input at reset) of each bit. External port pins may also request an interrupt to the CPU (maskable) when programmed to be an input.

Interrupt control

The device contains several interrupt sources. Each can be masked by setting a bit in a control register. Priority is resolved in software; all 'interrupt request' bits from the various sources are readable in a register. This register can be written to; writing a '1' clears the corresponding request bit, but writing a '0' has no effect. Individual interrupt control registers also exist within each of the functional blocks (HDLC controller, UART etc.). The registers described here provide a centralized and thus fast means of handling priorities. The various interrupt sources are permanently routed to the nIRQ (normal interrupts) and to the FIRQ (fast response) of the ARM7 CPU.

Timers / watchdog
2, 16 bit timer / counters and a (1-second) Watchdog timer are included. The timers support auto-pre-load timer interrupt generation, thus allowing interrupts to be generated at regular, programmable intervals. Timer 2 also support timer-capture functions (the source of which is selectable). This allows the timing of external events to be simplified (one common application being hook-flash detection on the analog ports).

Detailed Block Description

GCI frame formats:
The contents of one GCI frame is different whether it concerns an ISDN interface (U or S) or an analogue interface (SHPOTS). ISDN connections use two D-channel bits, which are used as normal signalling / control bits for SHPOTS.
The MTC-20280 however will always handle the U, S, and A interfaces in the same way: They are functionally identical and can be interchanged, the names U, S, and A being used for convenience only. Therefore, the two D and four C / I bits are always considered separately as for an ISDN connection.
The software is responsible for correct interpretation of the two formats, i.e. consider the two D bits as two additional CI bits.

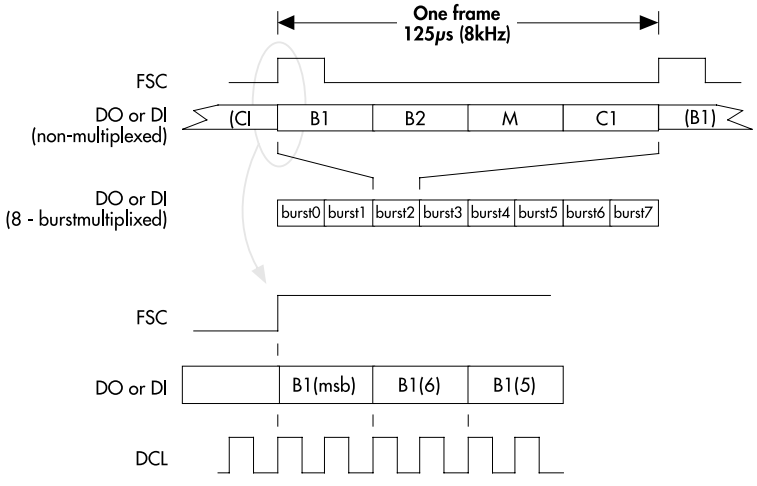


Figure 6 : GCI Frame Formats

The meaning of the six C / I bits for Alcatel Microelectronics' SHPOTS is described in the MTK-40131 data-sheet. The interpretation of these for the ISDN components MTC-2071 (single 4B3TU interface), MTC-20172 (S-interface), MTC-20276 and MTC-20277 (Integrated NT devices for 2B1Q and 4B3T respectively) is described in detail in the data-sheets for these products.

Serial communication is done MSB first; therefore the MSB of the data in parallel registers of the MTC-20280 that are related to GCI, will be sent / received as the first bit.

GCI VERSION	**B1 channel** (1 byte)	**B2 channel** (1 byte)	**Monitor channel** (1 byte)	**C/I channel** (1 byte)	
				Signalling and control bits	Monitor handshake bits
ISDN	B1 (8)	B2 (8)	M (8)	D (2) C/I (4)	A/E (2)
ANALOG	B1 (8)	B2 (8)	M (8)	C/I (6)	A/E (2)

In order to cope with the difference in convention between GCI and HDLC (LSB first) formats, the HDLC modules offer the possibility to do bit-reversal on the parallel data when the data are not to be HDLC formatted. This is selected by means of a user-programmable control bit.

The MTC-20280 supports multiplexed GCI channels: E.g. an 8-channel multiplexed mode, in which case the frame contents as described above, is sent eight times faster over the bit-serial I/O. According to the GCI specification, the first transmitted channel is called Channel0, the last one Channel7; these channels are referenced to as "Burst0" up to "Burst7" in the following sections. GCI frames of up to 3088 kbps (eight bursts x 32bit + 130 spare bits, all at 8kHz) are accepted if generated by an external GCI master. An external GCI master may also apply less than eight bursts, e.g. a 768kbps GCI frame of three bursts of 32 bits.

Parameter updating: timing and initial values

The following sections describe the functional building blocks and their parameters, which are stored in CPU-addressable registers in the memory space.

The default rule is that all parameters can be read or written at any time, and the new value that is written is immediately used. This is also valid for the GCI related parameters GCI_CPU_RX, GCI_SRC_BUR, GCI_ITx and GCI_MSCx (x=U,S,A).

However, for the following GCI related registers an exception is made on this default rule. The Source, CPU and Swap registers of the GCI router (i.e. all registers named "_SRC", "_CPU" and "_SWP) can be written at any time, but will only be updated /used based on the rate at which the RAM open/closed space will be switched (see architecture of GCI router). The rate is equivalent to the GCI frame rate, but the moment of switching coincides with the last byte of the last burst in the GCI frame (see next Figure). The value that is read is the value that is used in the current GCI frame I; therefore, the read value can be different from the last written value (if no update was done since it was last written).

The HDxTX registers are read only and show the value which is being transferred in the current GCI frame (see next Figure). The RX registers (i.e. all registers named "_RX") are read only and show the value which was received during the previous GCI frame (see next Figure).

For the RX register corresponding to the last byte of the last burst in the GCI frame, an exception must be made, as shown in the figure. The value should not be read during the time window starting the moment the RAM space is swapped until the start of a new GCI frame. (It is therefore recommended to read these registers soon after the frame-interrupt request).

Notice that after power reset, no write operations will be issued before the first GCI frame FSC is generated. This in order to allow correct initialisation of the MTC-20280.

Notice also that, after power reset, the default GCI clocks selected for U/S/A is the 'power down' mode. This means the FSC/DCL remains inactive low ('0'), but also the output GCI data stream remains inactive high ('1': idle). Thus, none of the uninitialised register values will be put onto the output stream, except under software control. The software has to do the appropriate initialisation before selecting it as source register.

When the GCI clocks are switched from one to another source (e.g. between Umaster and crystal based clocks on the A interface), the data could be unpredictable during the switch. This is because a re-synchronisation must take place unless specific measures are taken. The MTC-20280 GCI clock circuitry contains a digital phase-locked loop (DPLL) which allows timing differences to be tracked. See the section on GCI clock generation for details.

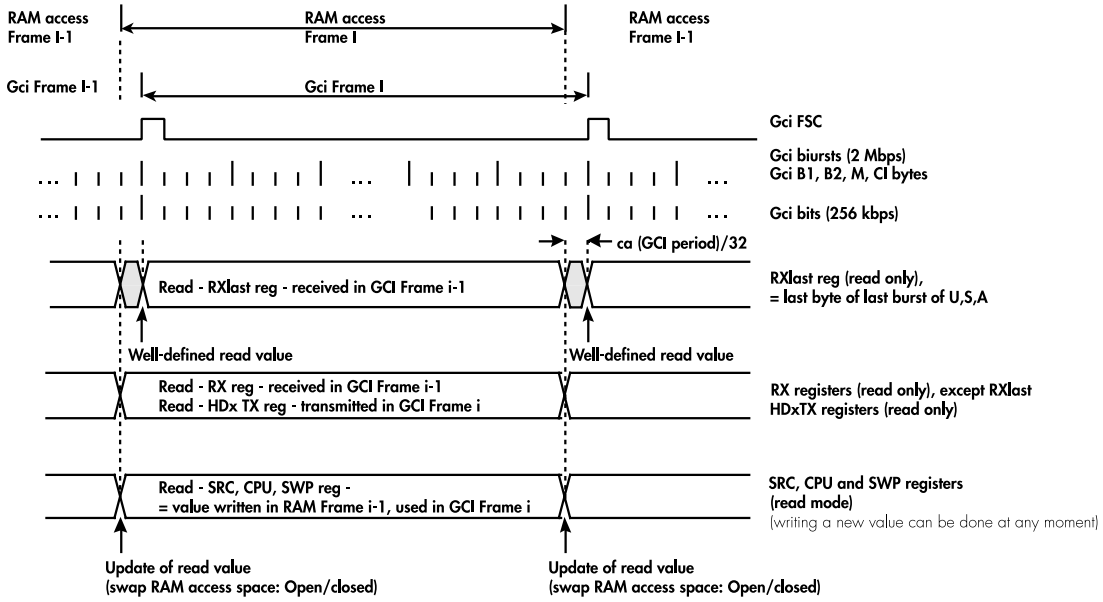


Figure 7 : Timing of GCI Control Register Updates

Handling D channel collisions

The GCI router and MTC-20280 architecture / parameters allow the control of whether the D-channel from S to U upstream is busy or not; it allows use of the output of an HDLC block as D-channel U-up when S is inactive. Details on this are described in the 'Programming' section.

Reset

The MTC-20280 has two reset pins: NRST (Functional reset) and NTRST (JTAG reset). Both active low reset inputs are independent and do not interact.

NRST: resets the integrated ARM processor (i.e. disables the ARM clock) and the functional MTC-20280 blocks, without affecting the JTAG related memories. (I.e. the test configuration register, down-loaded via the bit-serial JTAG interface is not reset. If set in a specific test mode, this mode will remain selected while resetting the ARM and functional blocks).

NTRST: resets the JTAG interface logic, including the test configuration register. This puts the MTC-20280 in non-test, functional mode. Note that the NTRST input cell has an integrated pull-down, such that the JTAG logic is reset by default, without requiring a connection to this pin on the PCB.

Note also that the NRST pin is made bidirectional, such that it can indicate when the watchdog resets the MTC-20280.

GCI Router Architecture

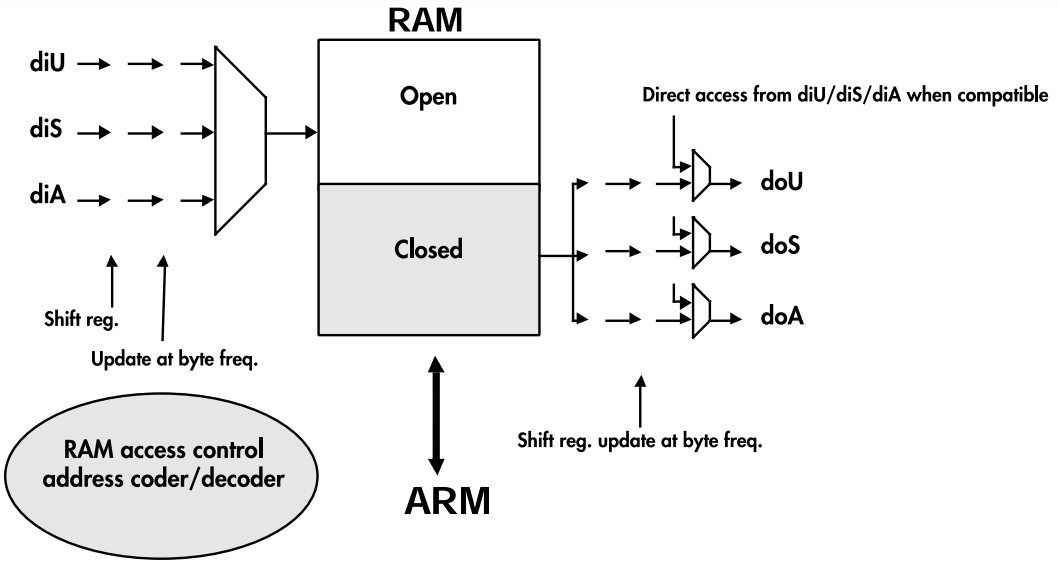


Figure 8 : GCI Router Architecture

The core of GCI Router is made of a dual port RAM. One port is dedicated to the GCI interfaces, the other one to the processor access.

The RAM is split into two spaces:

Open: space where the data can be modified, used to store the incoming GCI frames.

Closed: space where the data are held for one GCI frame, used to store the GCI frames to be sent.

This architecture allows the reception and transmission of up to eight bursts per GCI interfaces (max 8x3x4 bytes).
Should any GCI interface work faster than 2048 kbps (e.g. 3088 kbps, the highest GCI rate accepted by MTC-20280), the extra bits received by the MTC-20280 will not be stored in the RAM. The corresponding output stream generated by the MTC-20280 will contain idle spare bits (i.e. value = '1' according to GCI).

Multiplexer

The CPU can program an automatic routing of any channel from any burst through the GCI buses, from the CPU addressable registers or from one of the 3 HDLC formatters.

The source of each channel output is controlled by the value stored in the source control registers, as shown in the next register table. The registers can be written by the ARM: if a new value is written to the register, it will only be used from the next GCI frame onwards. However, due to the RAM architecture, a valve which must stay constant indefinately must be written in two concecutive frames. Before reading such a SRC register, the burst number must first be specified by writing into register GCI_SRC_BUR. This is because up to eight sources can be stored in a SRC register, one for each possible burst (see examples further on).

In addition, "swapping" can also be programmed between the channels B1 and B2 via the so-called SWP registers. (This allows re-allocation of the B1 and B2 channels, should the external device not support this function).

Name	Address		Function
	word	byte	
GCI_SRC_BUR	7F	200	Burst selection for reading Source control registers: [2:0] = target burst (0 to 7)
U_B1_SRC	80		Source control of B1 U-up channels routing : [2:0] = target burst (0 to 7)
			[4:3] = source line 0 = U-down channel 1 = S-up channel 2 = A-up channel 3 = Extension [7:5] = if source line = U or S or A : source burst (0 to 7) if source line = Extension: : one of following possible sources: 0 = IDLE 1 = CPU register 2 = HDLC1 3 = HDLC2 4 = HDLC3
U_B2_SRC	81	204	Source control of B2 U-up channels routing description: idem
U_D_SRC	82	208	Source control of D U-up channels routing description: idem
U_CI_SRC (*)	83	20C	Source control of C/I U-up channels routing description: idem, except for HDLC source (*)
U_M_SRC (*)	84	210	Source control of M U-up channels routing description: idem, except for HDLC source (*)
U_AE_SRC(*)	85	214	Source control of A/E U-up channels routing description: idem, except for HDLC source (*)
S_B1_SRC	86	218	Source control of B1 S-down channels routing description: idem
S_B2_SRC	87	21C	Source control of B2 S-down channels routing description: idem
S_D_SRC	88	220	Source control of D S-down channels routing description: idem
S_CI_SRC (*)	89	224	Source control of C/I S-down channels routing description: idem, except for HDLC source (*)
S_M_SRC (*)	8A	228	Source control of M S-down channels routing description: idem, except for HDLC source (*)
S_AE_SRC(*)	8B	22C	Source control of A/E S-down channels routing description: idem, except for HDLC source (*)
A_B1_SRC	8C	230	Source control of B1 A-down channels routing description: idem
A_B2_SRC	8D	234	Source control of B2 A-down channels routing description: idem
A_D_SRC	8E	238	Source control of D A-down channels routing description: idem
A_CI_SRC (*)	8F	23C	Source control of C/I A-down channels routing description: idem, except for HDLC source (*)
A_M_SRC (*)	B0	2C0	Source control of M A-down channels routing description: idem, except for HDLC source (*)
A_AE_SRC(*)	B1	2C4	Source control of A/E A-down channels routing description: idem, except for HDLC source (*)
U_B1_SWP	B2	2C8	bit[i] = 0: no swap; B1-burst i U-up channel will be routed with the B1 channel of the selected source (U,S,A only) = 1: swap; B1-burst i U-up channel will be routed with the B2 channel of the selected source (U,S,A only)
U_B2_SWP	B3	2CC	bit[i] = 0: no swap; B2-burst i U-up channel will be routed with the B2 channel of the selected source (U,S,A only) = 1: swap; B2-burst i U-up channel will be routed with the B1 channel of the selected source (U,S,A only)

Name	Address		Function	
	word	byte		
S_B1_SWP	B4	2D0	bit[i] = 0: no swap;	B1-burst i S-down channel will be routed with the B1 channel of the selected source (U,S,A only)
			= 1: swap;	B1-burst i S-down channel will be routed with the B2 channel of the selected source (U,S,A only)
S_B2_SWP	B5	2D4	bit[i] = 0: no swap;	B2-burst i S-down channel will be routed with the B2 channel of the selected source (U,S,A only)
			= 1: swap;	B2-burst i S-down channel will be routed with the B1 channel of the selected source (U,S,A only)
A_B1_SWP	B6	2D8	bit[i] = 0: no swap;	B1-burst i A-down channel will be routed with the B1 channel of the selected source (U,S,A only)
			= 1: swap;	B1-burst i A-down channel will be routed with the B2 channel of the selected source (U,S,A only)
A_B2_SWP	B7	2DC	bit[i] = 0: no swap;	B2-burst i A-down channel will be routed with the B2 channel of the selected source (U,S,A only)
			= 1: swap;	B2-burst i A-down channel will be routed with the B1 channel of the selected source (U,S,A only)

Notes:
All register contents are undefined after reset, which means that the user software must initialise all the register values before using them. Note however that the default source of all GCI channels is the IDLE source, such that all output streams will be continuously '1' (idle).

All registers can be read and written: Write operation: the new value will be effective from the next GCI frame on. Read operation: value used for the current GCI frame is read The HDLC source selections are not applicable for the CI, M and AE channels (see registers marked (*)); the bits are unspecified if the HDLC would still be selected as source. Therefore, it is advised not to do so.

For an 'Analog' GCI frame, the D and CI channels form one entity but the MTC-20280 handles them separately; therefore, the user software should use the same selection for both D and CI channels. For the data routed from one GCI interface to another one, a delay of one GCI frame is inserted.

If the target burst number is set higher than the number of bursts which may be transmitted on a GCI output stream, it will be neglected. Conversely, for a source burst number higher than the number of bursts in the GCI input stream, the content of the byte will be unspecified. In case of a non-multiplexed GCI stream, the source and target burst numbers must be set to '0'.

Examples:
If one wants to connect the destination "U / burst3 / B1" with the source "S / burst7 / B1", then make

- U_B1_SRC[7..0] = "111 01 011" (source burst=7 ; source=S ; target burst=3), and
- U_B1_SWP[7..0] = "xxxx 0xxx" (bit3 set to '0' in order not to swap the source of U/B1: B1)

If one wants to connect the destination
- "U / burst3 / B1" with the source "S / burst7 / B2", then make
- U_B1_SRC[7..0] = "111 01 011" (source burst=7 ; source=S ; target burst=3), and
- U_B1_SWP[7..0] = "xxxx 1xxx" (bit3 set to '1' in order to swap the source of U/B1: B2)

Note that the SWAP register values are only used if the corresponding source is U, S, or A (upstream or downstream) and is not an HDLC controller or IDLE, for example.

Reading the SRC value shows the connection that is selected for the current GCI Frame. Because eight values can be specified at the same time for each SRC register (e.g. U_B1_SRC, one value per target burst), reading U_B1_SRC necessitates first writing to register GCI_SRC_BUR to choose one target burst:

- Write U_B1_SRC[7..0] = "110 01 011" (source burst=6 ; source=S ; target burst=3)
- Write U_B1_SRC[7..0] = "111 00 010" (source burst=7 ; source=U ; target burst=2)
- Write U_B1_SRC[7..0] = "101 10 001" (source burst=5 ; source=A ; target burst=1)
- Write GCI_SRC_BUR[2..0] = "011" (specify target burst=2 for reading)
- Read U_B1_SRC[7..0] = "111 00 010" (source burst=7 ; source=U ; target burst=2)
- Write GCI_SRC_BUR[2..0] = "011" (specify target burst=3 for reading)
- Read U_B1_SRC[7..0] = "110 01 011" (source burst=6 ; source=S ; target burst=3)

CPU registers (ARM "→" GCI)

Through the CPU registers, the ARM can send data directly to any GCI channel.

Register table:

Name	Address		Function
	word	byte	
Ui_B1_CPU	60	180	CPU source value register for U-up / B1, burst i i is defined by the register GCI_CPU_RX
Ui_B2_CPU	61	184	CPU source value register for U-up / B2, burst i
Ui_M_CPU	62	188	CPU source value register for U-up / M, burst i
Ui_E_CPU	63	18C	CPU source value register for [7:6] = U-up / D, burst i [5:2] = U-up / CI, burst i [1:0] = U-up / AE, burst i
Si_B1_CPU	64	190	CPU source value register for S-down / B1, burst i
Si_B2_CPU	65	194	CPU source value register for S-down / B2, burst i
Si_M_CPU	66	198	CPU source value register for S-down / M, burst i
Si_E_CPU	67	19C	CPU source value register for [7:6] = S-down / D, burst i [5:2] = S-down / CI, burst i [1:0] = S-down / AE, burst i
Ai_B1_CPU	68	1A0	CPU source value register for A-down / B1, burst i
Ai_B2_CPU	69	1A4	CPU source value register for A-down / B2, burst i
Ai_M_CPU	6A	1A8	CPU source value register for A-down / M, burst i
Ai_E_CPU	6B	1AC	CPU source value register for [7:6] = A-down / D, burst i [5:2] = A-down / CI, burst i [1:0] = A-down / AE, burst i
GCI_CPU_RX	78	1EO	Specifies the target burst for line U,S or A used when accessing the CPU and RX registers: [1:0] = target line; 0 = U, 1 = S, 2 = A [4:2] = CPU target burst (0 to 7) [7:5] = RX target burst (0 to 7)

Notes:

All registers are undefined after reset (idem as for GCI multiplexing registers)
The register GCI_CPU_RX can define and store three commands in parallel: one for each interface (U, S, and A). The last stored command for each interface will be used to select the corresponding CPU and TX register of that source.
Example:
Initialisation: select for S and U:

- GCI_CPU_RX = " 000 010 01 " ; S: CPU-burst=2 , RX-burst=0
- GCI_CPU_RX = " 011 110 00 " ; U: CPU-burst=6 , RX-burst=3

use CPU and RX registers:
- Si_B2_CPU = Ui_B1_RX; means CPU / S_B2 / burst=2 gets data from RX / U_B1 / burst=3

re-select for S:
- GCI_CPU_RX = " 000 101 01 " ; S: CPU-burst=5 , RX-burst=0

use CPU and RX registers:
- Si_B2_CPU = Ui_B1_RX; means CPU / S_B2 / burst=5 gets data from RX / U_B1 / burst=3

All CPU registers can be read and written:
write operation: new value will be effective from the next GCI frame.
read operation: value used for the current GCI frame is read

NOTE: When a constant value needs to be used from a CPU register, the value should be written into the CPU register twice in two consecutive GCI frames. (This is an artifact of the architecture with two parallel RAM spaces – the constant value must be written into both RAM spaces. Failure to do so results in the output value alternating between the LAST output value and the new ('constant') value).

RX registers (GCI "→" ARM)

The CPU can read the GCI content on each interface (U, S, and A) at any time, with a delay of one frame:

Register table:

Name	Address word	Address byte	Function
Ui_B1_RX	6C	1B0	First byte of the U downstream GCI frame, burst i
Ui_B2_RX	6D	1B4	Second byte of the U downstream GCI frame, burst i
Ui_M_RX	6E	1B8	Third byte of the U downstream GCI frame, burst i
Ui_E_RX	6F	1BC	Fourth byte of the U downstream GCI frame, burst i i is defined by the register GCI_CPU_RX
Si_B1_RX	70	1C0	First byte of the S upstream GCI frame, burst i
Si_B2_RX	71	1C4	Second byte of the S upstream GCI frame, burst i
Si_M_RX	72	1C8	Third byte of the S upstream GCI frame, burst i
Si_E_RX	73	1CC	Fourth byte of the S upstream GCI frame, burst i i is defined by the register GCI_CPU_RX
Ai_B1_RX	74	1D0	First byte of the A upstream GCI frame, burst i
Ai_B2_RX	75	1D4	Second byte of the A upstream GCI frame, burst i
Ai_M_RX	76	1D8	Third byte of the A upstream GCI frame, burst i
Ai_E_RX	77	1DC	Fourth byte of the A upstream GCI frame, burst i i is defined by the register GCI_CPU_RX
GCI_CPU_RX	78	1E0	Specify the burst targeted by the ARM when writing to the CPU and RX registers: [1:0] = target line; 0 = U 1 = S 2 = A [4:2] = CPU target burst (0 to 7)

Notes:
All RX registers can only be read: The value received during the previous GCI frame is read.

D/CI Activity Detection

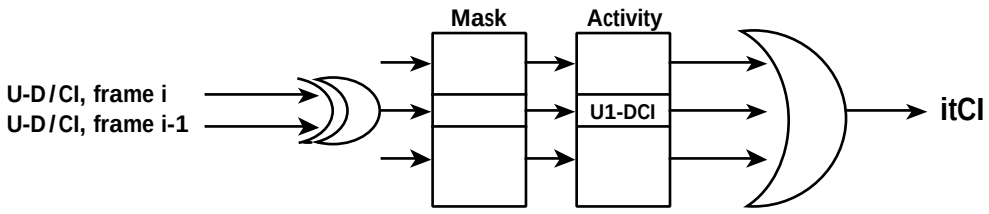


Figure 9 : C/I Bit Interrupt Architecture

The value of each bit of the six D/CI-bits of one GCI burst is compared to its value during the previously received GCI frame.

If at least one of the six bits of burst k changed, the corresponding burst-bit k=0..7 is set in the CI-activity-detection register, unless it is masked (by setting the corresponding 'mask bit' to 1).

There are three D/CI-activity-detection registers: one for each interface (U, S, and A), each eight bits wide (for the eight possible bursts per interface).

There is one mask register per D/CI-activity-detection register. After reset, the mask registers are set to 0xFF, all activity detection being masked.

If less than 8 bursts are present on a certain GCI interface, the activity bits corresponding to the non-existing bursts will remain inactive '0'; so no masking is required to prevent an incorrect activity detection.

From the moment that one of the 3x8 CI-activity bits is active, an interrupt request is generated.

The interrupt requests stay active until they are explicitly cleared by writing to the GCI_ITx registers.
The interrupt source itCI is internally routed to the interrupt handler (i.e. the raw, unmasked interrupt bit). It is also routed to the capture signal of Timer1, which allows the time between events to be measured (e.g. for detecting dial-pulses on a POTS interface).

Note that no debouncing is done on the activity detection. If necessary, this must be done in software.

Register table:

Name	Address		Function
	word	byte	
GCI_ITU	79	1E4	READ access: bit[i]=1 : CI activity detected on U-downstream, burst I WRITE access: bit[i]=1 : reset the interrupt detection on U-downstream, burst I
GCI_ITS	7A	1E8	READ access: bit[i]=1 : CI activity detected on S-upstream, burst I WRITE access: bit[i]=1 : reset the interrupt detection on S-upstream, burst i
GCI_ITA	7B	1EC	READ access: bit[i]=1 : CI activity detected on A-upstream, burst i WRITE access: bit[i]=1 : reset the interrupt detection on A-upstream, burst i
GCI_MSKU	7C	1F0	bit[i]=1 : mask CI activity detection on U, burst i
GCI_MSKS	7D	1F4	bit[i]=1 : mask CI activity detection on S, burst i
GCI_MSKA	7E	1F8	bit[i]=1 : mask CI activity detection on A, burst i

GCI Clock Generation

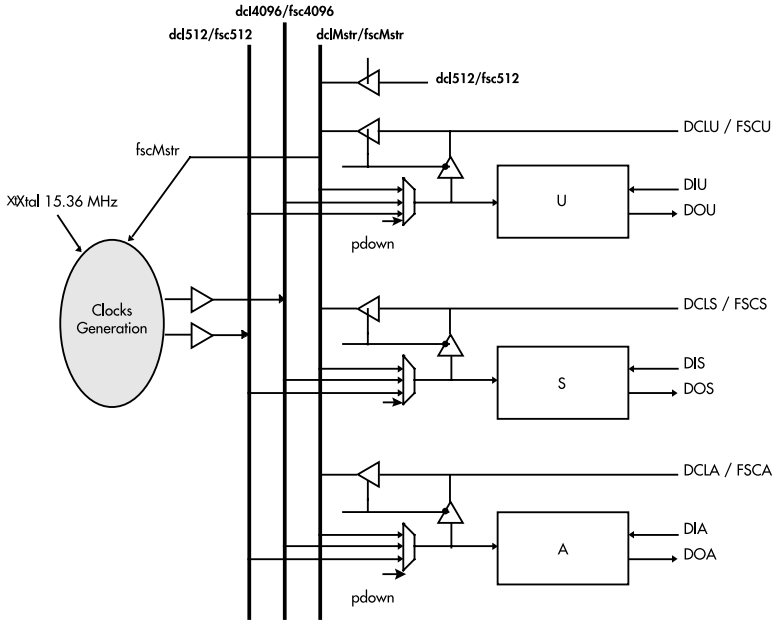


Figure 10 : GCI Clock Generation

Each GCI interface can work at a different speed (=clock domain fsc / dcl), but in all cases each (U,S,A) is synchronised via its FSC signal on one and only one GCI Master FSC (fscMstr). The selection of fscMstr as well as the clock domain to be used for U,S,A is under software control.

As each interface can be selected to be a GCI master, and at most one interface can work as master and the others as slave, all FSC / DCL pins are bidirectional. After reset, all three pairs of pins are in input mode such that no conflict occurs on any of the interfaces with a possible external master device.

The Master FSC (fscMstr) can be selected from four different sources, and the corresponding DCL clock will also be used as an input to the MTC-20280:

- Crystal clocks = dcl512 / fsc512
- U GCI interface = DCLU / FSCU
- S GCI interface = DCLS / FSCS
- A GCI interface = DCLA / FSCA

The default source after reset is the internal crystal, because this source will always be present.

For each clock domain fsc / dcl of U,S,A , one can choose between four possible sources:

non-active: this is the default after reset, and means dcl / fsc of that interface remains inactive low and the GCI data output stream remains inactive high (idle code). Selection of the non-active source for the interface, will overrule any other source selected for the data stream (e.g. U_B1_SRC = HDLC1
will be overruled)

master (dclMstr / fscMstr): the dcl / fsc clocks of that interface will follow the dclMstr / fscMstr of the interface which is selected as master interface.

512k (dcl512 / fsc512): the dcl / fsc clocks are derived from the crystal (15.36 MHz), but synchronized (1)
with the master frame clock (fscMstr). Dcl = 512 kHz, 1 burst per frame.

4096k (dcl4096 / fsc4096): the dcl / fsc clocks are derived from the crystal (15.36 MHz), but synchronized with the master frame clock (fscMstr). Dcl = 4096 kHz, 8 burst per frame.

Note: When Xtal is selected as master, fsc is synchronised with a free running, internally generated crystal-based 8kHz FSC signal.

The source specified for that GCI interface which is selected as Master, becomes irrelevant: whatever source is specified, it will not be used such that no conflicts can arise.

With a crystal based master
(the MTC-20280 being GCI master), the MTC-20280 can only generate either non-multiplexed (one burst), or 8-burst multiplexed GCI frame formats. However, when an external master is specified, the MTC-20280 will follow the format of the external master: e.g. a 5-burst mode, or an 8-burst mode with extra spare bits. So the MTC-20280, when in slave mode, is fully compatible with any other GCI compatible device.

Bit-rates of up to 3088 kbps may be applied as master; this corresponds to a maximum of 8 burst of 32 bits, followed by 130 spare bits (see GCI specification). Note that should spare bits occur in the input data stream, these will not be stored for possible processing, nor be routed through to another GCI output data stream (e.g. from DIU to DOS). Incoming spare bits are neglected, and outgoing spare bits are always set idle '1'.

Hardware Implementation of the GCI Clock Multiplexing

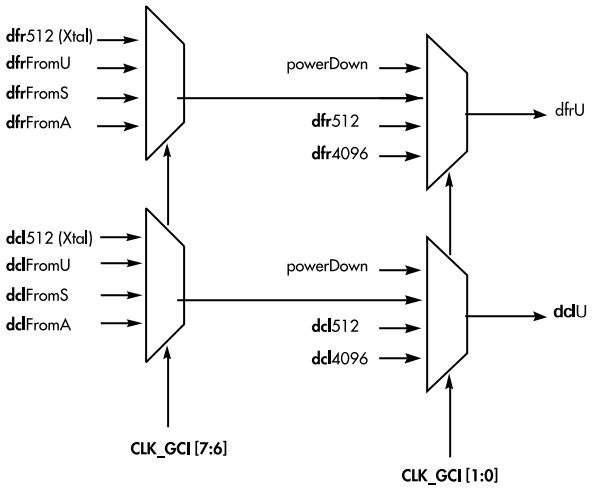


Figure 11 : Multiplexing of the GCI Clocks (U Interface Example)

The control of the multiplexers comes directly (asynchronously) from the CPU register CLK_GCI; the application software must take care when it modifies the multiplexing. Typically, the control will be set once at power-up initialisation because it's mainly dependent of the devices connected to the GCI lines. The dcl and dfr signals have the same source.

Data frame tracking

The generated dfr signals (dfr512 and dfr4096) are slaved to the dfr signal that has been chosen as master. An on-chip digital PLL (DPLL) can make a correction of maximum +/- 1.7% per frame; so a maximum of 30 frames (=50%/1.7%) can be needed when changing the dfr master.
The frequencies of the data clocks (dlc512 and dcl4096) are adapted in the same way.

The DPLL adds/substacts a maximum of 32 pluses of the 15.36 MHz clock per frame (= 32/1920=1.7% : 15.36 MHz = ± 65 ns). The DPLL works with a hysteresis of 65 ns; a phase shift of +/-65 ns or more will caused a correction. A flag bit reports the status of the DPLL (frames synchro. /not synchro. See the CLK_3 register description). If no frame clock is present on the selected dfr master, no tracking will be applied, so the generated dfr's will be free running.
Application example: For the S interface, use the internal clocks (dfr512/dcl512) and the frame clock coming from the U as dfr master. As soon as no activity occurs on the U frame, the S frame is free running (based on the crystal frequency). Immediately the U activated, the DPLL begins to recover an eventual frame phase delay between U and S.

Register table:

Name	Address		Function
CLK_GCI	90	240	bit[5:0]: GCI DCL clock selection per interface U,S,A: bit[1:0] = for the U GCI router bit[3:2] = for the S GCI router bit[5:4] = for the A GCI router 0 = non-active (default at reset) 1 = master : dclMstr / fscMstr 2 = 512k: dcl512 / fsc512 3 = 4096k: dcl4096 / fsc4096 bit[7:6] : GCI Master FSC selection (fscMstr) 0 = Crystal oscillator (default at reset) 1 = U 2 = S 3 = A
CHIP_GCI_L CHIP_GCI_M	02 03	8 C	Bits per GCI Frame on GCI master interface (Read only): result of auto-detection of the mode of the GCI master interface this 16-bit word stored in upper (_M) and lower byte (_L) value corresponds to the number of bits detected in one GCI frame: e.g. 8-burst 2048 kbps : CHIP_GCI_[M/L] = 256 = ox 01 00 = [1 / 0] 3-burst mode : CHIP_GCI_[M/L] = 96 = ox 00 60 = [0 / 96] 8-burst 3088 kbps : CHIP_GCI_[M/L] = 386 = ox 01 82 = [0/130]
CLK_3	93	24c	bit [0] : DPLL status (read only) 0 = internal gci clocks not synchronized with the extern reference; dpll is tracking 1 = internal gci clocks synchronized bit [1] : UART clock disable bit [3:2]: APB clock division factor 0 = 15.36 Mhz/4 (max.speed) 1 = 15.36 Mhz/8 2 = 15.36 Mhz/16 3 = 15.36 Mhz/32

Note:
The 4096 kHz clock generated from the crystal does not have a perfect 50% duty cycle clock. (The ratio 15360 to 4096 is not an integer value).

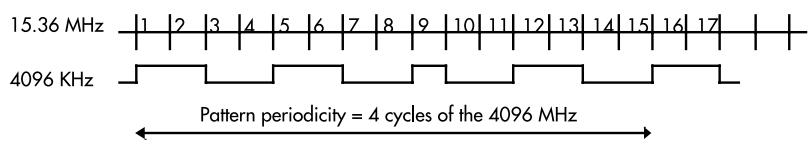


Figure 12 : Pattern Periodicity of the 4096 kHz GCI Data Clock

HDLC Formatter

Description

Three identical HDLC formatters are provided. They can be routed to any B1, B2, or D channels for any burst of any U, S, or A interface. Each HDLC formatter works as a single-channel HDLC controller with separate send and receive FIFO pools. It is designed to work in OS1 Layer2 (Data Link layer) applications, such as a LAP-D or LAP-B processor.

Features:

- Flag generation and detection
- Abort generation and checking
- CRC generation and checking
- Zero insertion and deletion
- Receive address comparison
- 1 broadcast TEI register
- 2 programmable address matching registers
- 2 programmable "wildcard" registers
- Transmit address of packets can be set from register or data stream
- 64 byte FIFOs in both directions
- Transparent mode, where the FIFOs can be used to buffer data transfers without HDLC formatting.
- Data bit-reversal possible (lsb ´ msb) in order to cope with the different formatting between HDLC (LSB first) and non-HDLC, GCI formatted data (MSB first).
- Interruption generation

HDLC protocol

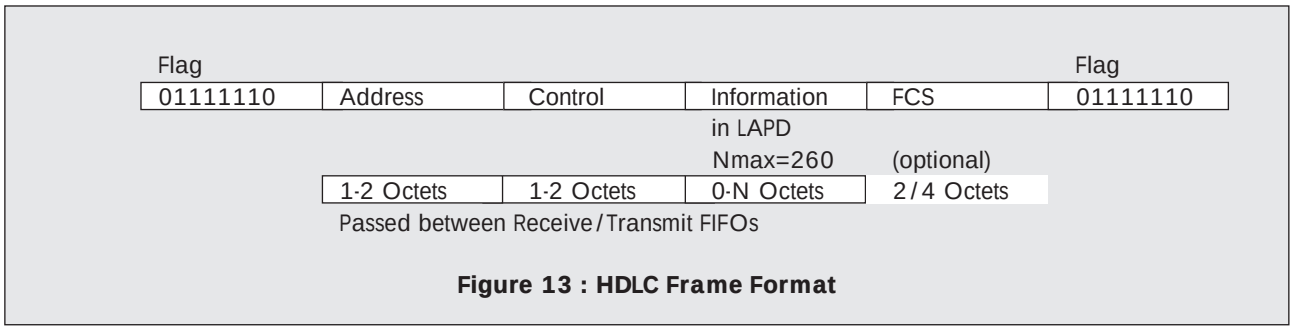


Figure 13 : HDLC Frame Format

HDLC (High Level Data Link Control) is a bit-oriented, synchronous serial protocol used in data communications systems. Both LAPD (Link Access Protocol on D-channel) and LAPB (Link Access Protocol Balanced) are based on HDLC, differing only in frame content.

The HDLC block transmits and receives data in frames. The start and end of frames are marked by a unique bit pattern called a flag. The data between the start and end flags consists of an address field, control field, information field and a Frame Check Sequence (FCS) field.

Figure 13 shows the HDLC frame format.

Framing

A flag is the unique bit-pattern '01111110' (7E hex), and marks both the start and the end of the frame. Flags are generated internally, and the HDLC block automatically appends start and end flags on frame transmission. Flags are searched for on a bit by bit basis, and can be recognised at any point in the received bit stream.
Flags are not transferred to or from the HDLC block.

Addressing

The frame address is contained in the first field following the start flag. This can be either one or two octets long, and is used to distinguish the various network devices from one another. Together with optional address-matching circuit, the HDLC block can search the complete address field of incoming frames, selecting only those frames addressed to it. The address field is transferred to and from the HDLC block.

Control
A control field follows the address field. This can be either one or two octets long, and is used to transfer commands and responses between Layer 2 entities and the network. The HDLC block does not operate on this field, transferring it transparently.

Information
The information field contains the data to be transmitted, and may be null. This field is not necessarily an integer number of octets long. If the last few bits of the information field do not completely fill the last octet, then that octet is padded with zeros before being transferred to the Receive FIFO as a complete byte. The device will only transmit frames with an integer number of octets (before zero insertion).

Error checking
The Frame Check Sequence (FCS) field is contained in the last two octets before the end flag in a frame. The field is computed using a Cyclic Redundancy Check (CRC) polynomial. This is used to perform error detection on the address, control, and information fields. The standard CRC-CCITT polynomial is used in both the receive and transmit directions:

$X16 + X12 + X5 + 1$

The HDLC block also provides support for a non-standard 32 bit CRC (CRC-32), which may be used instead of the standard CCITT-CRC. The polynomial for this is:
$X32 + X26 + X23 + X22 + X16 + X12 + X11 + X10 + X8 + X7 + X5 + X4 + X2 + X + 1$

CRC generation and checking is performed automatically by the HDLC block. On transmit, the CRC generator is initialised with FFFF hex. The CRC is computed serially on the address, control, and information fields, and is then complemented before being transmitted MSB first.

In the receive direction the CRC checker is initialised with FFFF hex on receipt of a valid start flag. A CRC is performed on all bits between the start and end flags, including the transmitted CRC field, and the result is compared to 1D0F hex (or C704DD7B hex for CRC-32). The frame is transferred to the receiver FIFO, regardless of whether the CRC checker detected an error or not. The CRC field can also be transferred to the receiver FIFO if required.

Zero insertion / deletion
Zero insertion and deletion is performed by the HDLC block to prevent the start and end flags from being imitated by the data. The transmitter section inserts a zero after any succession of five 1's within a frame (i.e. between start and end flags). The receiver section automatically deletes all 0's inserted by the transmitter.

Inter-frame time fill
An inter-frame time fill condition occurs when the HDLC block has no frames to transmit. The device can be configured to either transmit an idle stream or flag characters during this period. The idle stream consists of binary 1's in the LAPD protocol: the number of 1's can be less than a full octet. The LAPB protocol, on the other hand, specifies flag characters to be inserted between frames.

Frame abort
Transmission of data frames can be prematurely cancelled by use of an abort character. The transmitter aborts a frame by sending the abort character '11111110' (FE hex). The receiver interprets this character as an abort, and begins the search for a new frame.

How to generate a Tx Frame Abort:
This can be done in one of the following 3 ways:
- disable the TX before the 'end-of-frame' (indicated by writing the 'last-byte-indication' to the TX-fifo)
- clear the TX-fifo before the 'end-of-frame'
- let the TX-fifo run empty before the 'end-of-frame'

HDLC control registers
This section details the programmable registers accessible to the CPU. These registers either control the operation

of the HDLC formatter, or report its status. HDx parameters in the next table refer to three parameters, x

referring to any of the three HDLC formatters.

Name	Address		Function
	word	byte	
HDx-SRC			Source data for the receive path: [1:0]: line selection
x=1	40	100	0 = U
x=2	41	104	1 = S
x=3	42	108	2 = A
			3 = / (unspecified; not to be used)
			[4:2]: burst selection (0 to 7)
			[6:5]: channel selection
			0 = B1
			1 = B2
			2 = D
			3 = / (unspecified; not to be used)
HD_BREV	43	10C	Bit-reverse data byte (lsb ´ msb) read / written from / to the HDLC FIFOs bit[0] = 1 : bit-reverse data for HDCL1 formatter bit[1] = 1 : bit-reverse data for HDLC2 formatter bit[2] = 1 : bit-reverse data for HDLC3 formatter bit[4] = 1 : bit reversed per block of 2 for D channel transmission through HDLC1 in transparent mode bit[5] = 1 : bit reversed per block of 2 for D channel transmission through HDLC2 in transparent mode bit[6] = 1 : bit reversed per block of 2 for D channel transmission through HDLC3 in transparent mode
HDx-TX			HDLC packet ready to be transmitted in the current frame
x=1	44	110	(read only register; the register is updated by and when the HDLC TX-fifo
x=2	45	114	is active, transmitting data)
x=3	46	118	
HDx-MODE			HDLC Mode Register (MODE) This register controls the formatter configuration
x=1	10	40	= [0, HEN, TXE, RXE, CR32, ITF, FLS, TIC]
x=2	20	80	HEN : HDLC Protocol Enable
x=3	30	C0	1 : HDLC protocol enabled (data sent LSB first)
			0 : HDLC protocol disabled (data sent LSB first)
			TXE : Transmitter Enable
			1 : Transmitter activated
			0 : Transmitter deactivated
			RXE : Receiver Enable
			1 : Receiver activated
			0 : Receiver deactivated
			CR32 : 32 bit CRC Enable
			1 : Enable 32 bit CRC (non-standard)
			0 : Disable 32 bit CRC
			ITF : Inter-frame Time Fill
			1 : Continuous 1's between frames
			0 : Flag characters between frames
			FLS : Flag sharing
			1 : Transmit one flag between frames
			0 : Transmit two flags between frames

Name	Address		Function
	word	byte	
			TIC : Transmit Incorrect CRC 1 : Transmit an incorrect CRC field value 0 : Transmit correct CRC field value
HDx-EMODE x=1 x=2 x=3	 11 21 31	 44 84 C4	HDLC Extended Mode Register (EMODE) This register controls the configuration of the extended HDLC modes. = [LPB, CPT, TAIE, RAF1, RAF0, MFLE, MFL1, MFL0] LPB : Loop-back (at parallel ARM-IO side, NOT at bit-serial GCI side) 1 : Transmit – receive looped back 0 : Transmit – receive not looped back CPT : CRC Pass Through 1 : Pass CRC bytes into the data stream 0 : Do not pass CRC bytes into data stream TAIE : Transmit Address Insertion Enable 1 : Transmit address octets sourced from TA1/2 0 : Transmit address octets sourced from data RAF[1:0] : Receive Address Filter 00 : No filter on receive addresses 01 : Frame accepted if first address octet corresponds to either RA1/RAW1 or RA2/RAW2 10 : Frame accepted if second address octet corresponds to either RA1/RAW1 or RA2/RAW2 11 : Frame accepted if first address octet corresponds to RA1/RAW1 and second address octet corresponds to RA2/RAW2 MFLE : Minimum Frame Length Check Enable 1 : Minimum frame length check enabled 0 : Minimum frame length check disabled MFL[1:0] : Minimum Frame Length Check 00 : Only receive frames of 3 bytes or more 01 : Only receive frames of 4 bytes or more 10 : Only receive frames of 5 bytes or more 11 : Only receive frames of 6 bytes or more
HDx-CMD x=1 x=2 x=3	 12 22 32	 48 88 C8	HDLC Command Register (CMD) Commands for the formatter are written to this register. Writing a '1' to any of the bit locations will cause the appropriate action to take place. = [0, 0, 0, 0, TFT, TFC, RFC, RFB] TFT : Transmit Frame Terminator 1 : The next byte to be written to the Transmit FIFO is the last of the frame TFC : Transmit FIFO Clear 1 : Transmit FIFO is cleared RFC : Receive FIFO Clear 1 : Receive FIFO is cleared RFB : Receive Frame Abort 1 : Abort the receive frame 0 : Transmit two flags between frames

Name	Address		Function
	word	byte	
HDx-SSTAT			HDLC Serial Status Register (SSTAT) This register contains the current status of the Formatter receive and transmit serial functions.
x=1	13	4C	= [0, 0, 0, 0, SPG, RID, RFL, TIF]
x=2	23	8C	SPG : Serial Port Grant : this bit will have no influence if being reset, because
x=3	33	CC	the it is always granted by hardware construction: 1 : Serial port granted 0 : Serial port has not been granted RID : Receive Line Idle 1 : Receiving idle characters 0 : Frames or starting flags are being received RFL : Receiving Flag Characters 1 : Receiving flag characters 0 : Idle / frame data being received TIF : Transmit in Frame 1 : Transmitting frame data characters 0 : Transmitting idle / flag characters
HDx-FSTAT			HDLC Fifo Status Register (FSTAT) This register contains the current status of the formatter.
x=1	14	50	= [0, RSB, TLL, TFE, TOU, RLH, RFE, ROU]reset value = 32 hex
x=2	24	90	RSB : Receive Status Byte
x=3	34	D0	1 : Next byte on receive FIFO is a status byte 0 : Next byte on receive FIFO is a data byte TLL : Transmit FIFO Level is Low 1 : Transmit FIFO level is less than its threshold level 0 : Transmit FIFO level is greater than or equal to its threshold level TFE : Transmit FIFO is Full / Empty 1 : Transmit FIFO full if TLL=0, empty if TLL=1 0 : Transmit FIFO is neither full nor empty TOU : Transmit FIFO has Overrun / Underrun 1 : Transmit FIFO overrun if TLL=0, underrun if TLL=1 0 : Transmit FIFO has neither overrun nor underrun RLH : Receive FIFO Level is High 1 : Receive FIFO level is greater than or equal to its threshold level 0 : Receive FIFO level is less than its threshold level RFE : Receive FIFO is Full / Empty 1 : Receive FIFO full if RLH=1, empty if RLH=0 0 : The receive FIFO is neither full nor empty ROU : Receive FIFO has Overrun / Underrun 1 : Receive FIFO overrun if RLH=1, underrun if RLH=0 0 : Receive FIFO has neither overrun nor underrun threshold level = half of the FIFO depth
HDx-ISTAT			HDLC Interrupt Status Register (ISTAT) = [TXOK, TXERR, RXOK, RXERR, TXFL, TXFU, RXFH, RXFO]
x=1	15	54	(see HDLC Interrupt Mask Register (HDx_IMASK))
x=2	25	94	Holds the current interrupt status;
x=3	35	D4	Reading this register returns the same value that was read during the last read of HDx_ISERV.

Name	Address		Function
	word	byte	
HDx-ISRV x=1 x=2 x=3	 16 26 36	 58 98 D8	HDLC Interrupt Service Register (ISRV) = [TXOK, TXERR, RXOK, RXERR, TXFL, TXFU, RXFH, RXFO] (see HDLC Interrupt Mask Register (IMASK)) Reading this register : returns a value indicating all pending interrupts, and clears the interrupts (ISRV reset to 00hex); The returned value is stored in HDx-ISTAT for further reading.
HDx-IMASK x=1 x=2 x=3	 17 27 37	 5C 9C DC	HDLC Interrupt Mask Register (IMASK) = [TXOK, TXERR, RXOK, RXERR, TXFL, TXFU, RXFH, RXFO] TXOK : Transmit Frame OK 1 : Frame transmitted OK 0 : No interrupt TXERR : Transmit Frame Aborted 1 : Transmit frame aborted 0 : No interrupt RXOK : Receive Frame OK 1 : Frame received OK 0 : No interrupt RXERR : Receive Frame Error 1 : Frame aborted / received with CRC error 0 : No interrupt TXFL : Transmit FIFO low 1 : Transmit FIFO level has fallen below threshold 0 : No interrupt TXFU : Transmit FIFO Underrun 1 : Transmit FIFO has underrun 0 : No interrupt RXFH : Receive FIFO High 1 : Receive FIFO level has risen above threshold 0 : No interrupt RXFO : Receive FIFO Overrun 1 : Receive FIFO has underrun 0 : No interrupt
HDx-FIFO x=1 x=2 x=3	 18 28 38	 60 A0 E0	TX / RX Data Reading this register pops data off the Receive FIFO, from the current read address, and writing this register pushes data onto the Transmit FIFO. Note that writing a '1' to the TFT bit in the HDx_CMD register before writing the last byte to this register will terminate the last byte in the HDLC frame. Note that when changing to transparent mode either from reset, or during normal operation, the first received byte in the RX FIFO should be ignored. = [D7, D6, D5, D4, D3, D2, D1, D0]
HDx-RFBC x=1 x=2 x=3	 19 29 39	 64 A4 64	Receive Frame Byte Count (RFBC) The contents of this register are valid after an RXOK / RXERR interrupt. The value in this register corresponds to the number of receive frame bytes placed into the Receive FIFO. This value includes CRC bytes if CPT=1, but does not include the Receive Status Byte placed into the Receive FIFO immediately following the frame data. = [C7, C6, C5, C4, C3, C2, C1, C0] (modulo 256)

Name	Address		Function
	word	byte	
HDx-TA1			Transmit Address 1 (TA1) This register contains the first address octet for outgoing HDLC frames.
x=1	1A	68	This register value will only be used as the first address octet if the TAIE bit in
x=2	2A	A8	the HDx_EMODE register is set to '1'.
x=3	3A	68	= [T17, T16, T15, T14, T13, T12, T11, T10]
HDx-TA2			Transmit Address 2 (TA2) This register contains the second address octet for outgoing HDLC frames.
x=1	1B	6C	This register value will only be used as the second address octet if the TAIE bit
x=2	2B	AC	in the HDx_EMODE register is set to '1'.
x=3	3B	EC	= [T27, T26, T25, T24, T23, T22, T21, T20]
HDx-RAW1			Receive Address Wildcard 1 (RAW1) This register contains a 'wildcard' pattern for use by the receive address register RA1.
x=1	1C	70	Any bits set to '1' in this register will not be used in the comparison with an
x=2	2C	B0	address octet.
x=3	3C	F0	= [RW17, RW16, RW15, RW14, RW13, RW12, RW11, RW10]
HDx-RAW2			Receive Address Wildcard 2 (RAW2) This register contains a 'wildcard' pattern for use by the receive address register RA2.
x=1	1D	74	Any bits set to '1' in this register will not be used in the comparison with an
x=2	2D	B4	address octet.
x=3	3D	F4	= [RW27, RW26, RW25, RW24, RW23, RW22, RW21, RW20]
HDx-RA1			Receive Address 1 (RA1) This register contains a value to be used in the address matching circuitry for
x=1	1E	78	received HDLC frames. The RAF bits in the HDx_EMODE register control the
x=2	2E	B8	operation of the address matching circuitry.
x=3	3E	F8	= [RA17, RA16, RA15, RA14, RA13, RA12, RA11, RA10]
HDx-RA2			Receive Address 2 (RA2) This register contains a value to be used in the address matching circuitry for
x=1	1F	7C	received HDLC frames. The RAF bits in the HDx_EMODE register control the
x=2	2F	BC	operation of the address matching circuitry.
x=3	3F	FC	= [RA27, RA26, RA25, RA24, RA23, RA22, RA21, RA20]

Notes:

On the reception of a frame, a receive status byte is appended to the frame data in the FIFO. This status byte indicates how successfully the frame was received. The status byte can be identified in the receive data stream either by examining the RSB status bit of the HDx_FSTAT register, or by examining the RFBC register value after an RXOK/RXERR interrupt has occurred.

Receive Status Byte = [0 , 0 , 0 , RAB , CRC/ERR , PAD2 , PAD1 , PAD0]

RAB : Receive Abort
- 1 : Receive frame aborted
- 0 : Receive frame not aborted

CRC/ERR : Receive CRC Error
- 1 : Receive CRC Error
- 0 : No receive CRC Error

PAD[2:0] : 3 bit number indicating the number of padding bits added to the final receive character in the case of non octet aligned data.

Clock generation
A master clock oscillator, based on an external crystal of 15.36 MHz is provided. This can provide an output at the crystal frequency for use by the ISDN chip (INTT or INTQ). It therefore offers better than 100-ppm accuracy ; the external crystal is specified to 50-ppm accuracy.

Note: the oscillator pins may also be controlled from an external master clock. In that case, the external master clock is connected to the pin XTAL1, whereas pin XTAL2 is connected to GND.

One clock output (CKOUT) with a programmable division ratio from the 15.36 MHz input clock, is provided for use by any external peripheral function. It is user-programmable via the control register CLK_1 from 15.36 MHz in even integer division steps. (15.36 MHz, 15.36/2 MHz, 15.36/4 MHz, 15.36/6 MHz, ...). The clock output can also be disabled; when the clock is disabled, it remains constant high.
Typically, CKOUT can be used to drive the 15.36MHz input clock of the INTT/INTQ and therefore this will be the default start-up value. An MTC-20172 device in a TE application, for example, requires a 7.68MHz input clock; in this case, the user should set this parameter to divide the clock by two.

The processor clock (ARM clock) is also a programmable clock derived from the 15.36 MHz input clock. It is user-programmable via the control register CLK_2 from 15.36 MHz in even integer division steps. (15.36 MHz, 15.36/2 MHz, 15.36/4 MHz, 15.36/6 MHz, ...).
In addition, the ARM clock can completely be disabled, bringing the ARM in complete power-down mode.

CPU clock control protection
In order to avoid a disabling of the clocks by accident, and thus causing a lock-up, a four-bit word must be written in specific register fields (CLK_1[7..4] and CLK_2[7..4]).

CKOUT must be enabled by software by writing any value different from "1001" to CLK_1[7..4]; at the same time the value written in CLK_1[3..0] defines the frequency used at restart.

For the ARM clock, this is slightly different. It is disabled under software control by writing the appropriate value to register CLK_2; this also defines the start-up frequency. However, enabling is done by an activity detection circuit, triggering the ARM interrupt (FIQ or IRQ). Therefore, before going into power-down, the software must take care that the appropriate interrupt input source is enabled (not masked); otherwise only a power reset will allow to start-up the ARM again. The bits CLK_2[7..4] will be reset automatically by the hardware activity detection.

Notice that during ARM power-down, all other hardware units (HDLC, UART,..) and the GCI interfaces will keep running.

After reset, the clock CKOUT and the ARM clock are set to the maximum frequency and are not disabled:

CKOUT = 15.36 MHz (CLK_1=0)
ARM clock = 15.36 MHz (CLK_2=0)

Furthermore, the clock generator will generate an input clock for the integrated UART block. This clock will be derived from the 15.36 MHz input clock and will have a fixed frequency of 3.6864 MHz. The UART can generate different baud rates based on that input clock.

Register table:

Name	Address		Function
	word	byte	
CLK_1 [3..0] CLK_1 [7..4]	91	244	Integer representing the CKOUT output clock frequency CKOUT = 15.36 MHz / (2 * CLK_1) Remarks: CLK_1[3..0] = 0 is translated into a division by one the max division factor is CLK_1[3..1]=15 CKOUT can be disabled by setting bits CLK_1[7..4] == "1001"

Name	Address		Function
	word	byte	
CLK_2 [3..0] CLK_2 [7..4]	92	248	Integer representing the ARM clock frequency ARM clock = 15.36 MHz / (2 * CK2) Remarks: CLK_2[3..0] = 0 is translated into a division by one the max division factor is CLK_2[3..0]=15 CLK2 can be disabled by setting bits CLK_2[7..4] == "1001"
CLK_3	93	24C	TBD bit [0] : DPLL status (read only) 0 = internal gci clocks not synchronized with the extern reference; dpll is tracking 1 = internal gci clocks synchronized bit [1] : UART clock disable bit [3:2]: APB clock division factor 0 = 15.36 Mhz/4 (max.speed) 1 = 15.36 Mhz/8 2 = 15.36 Mhz/16 3 = 15.36 Mhz/32

The ARM7TDMI and its interfaces

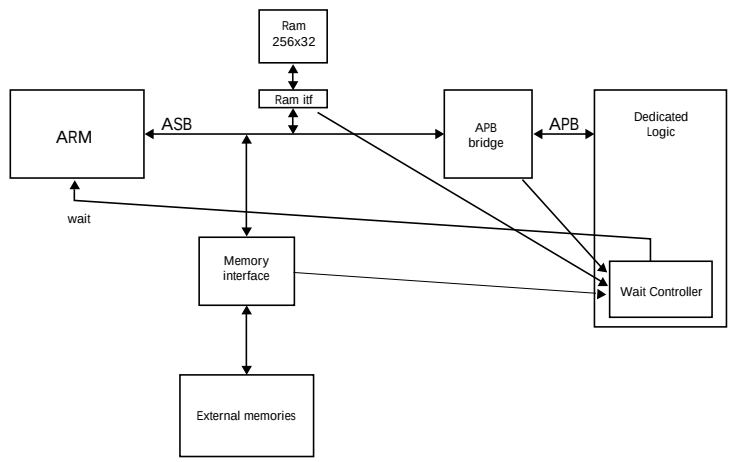


Figure 14 : The ARM7TDMI Processor Core and its Interfaces

ASB: Advanced System BusAsbClock: i.e. ARM clock CLK2 , (user programmable frequency)
APB: Advanced Peripheral Bus
ApbClock : fixed to Xtal/4

The ARM processor is configured to operate in 'Little-Endian' mode when treating the bytes in memory. The ARM core hardware decides whether an 8-bit (byte b_0), 16-bit (bytes = b_1, b_0 with b_0=LSB byte) or 32-bit word (bytes b_3, b_2, b_1, b_0 with b_0=LSB byte) is to be accessed. Dedicated Logic registers are 8-bit wide, so the ARM will only request for a 8-bit access and the APBBridge will only support Single Byte access. The physical address of each hardware register lies on a word boundary (NB. ARM Ltd. has chosen to call a 32-bit entity a 'word'. This convention is retained in this document).
The External Memory interface can be set to operate in one of four possible

modes. Depending on the mode, 16-bit and 32-bit accesses will be automatically transformed into one or two consecutive Two Byte accesses, or two or four consecutive Single Byte accesses respectively. The internal memory allows direct access to 1-, 2- or 4-byte words.

APB bridge

The APB clock is fixed to 15.36 MHz / 4 = 3.84 MHz for power optimization. However, depending on the ASB clock (= processor clock), the APB clock will be modulated in order to minimize the cycles needed for the read and write operations, thus maximizing processor throughput. Processor wait cycles will only be introduced by the hardware controller according to the following rules:

READ operations: no wait cycle
WRITE operations: two ASB cycles are needed; wait cycles will be introduced depending on an internal finite state-machine. If the previous WRITE operation is completed, no wait cycle is needed. One wait cycle will be introduced to wait until the end of the previous operation if needed. (This situation seldom occurs in practice, due to the processor executing an intermediate READ cycle to fetch the next instruction in most cases).

On chip memory

A RAM of 1kbyte with zero wait-state access is foreseen on chip. Read access is performed in one single cycle, for 8-bit, 16-bit as 32-bit word accesses. Writing 32 bit wide words is always done in one cycle. Write accesses of 8-bit and 16-bit words require at most 2 cycles, due to the 2 cycle read-modify-write sequence that is automatically generated by the internal memory interface. In order to reduce the processor wait cycles, the FSM will only introduce a wait cycle if the previous internal memory access operation is not fully completed (this is identical to the APB bridge case). This can only occur when a 8- or 16-bit write access is scheduled just after another 8- or 16-bit write access to the same on chip memory. (Mostly, the ARM will not execute two write operations successively, but will fetch a new instruction in the mean time).

External memory interface

The external memory interface consist of:

- A[18:1] and MC7 : 19 bits address bus
- DQ[15:0]: 16 bits data bus allowing SingleByte or TwoByte transfers; the MSB of the data bus DQ[15] may have a special function depending on the selected MemoryAccess configuration
- NWR,NOE: two control signals with fixed function
- MC7..0 : 8 Memory control output signals, of which the function depends on the selected MemoryAccess mode (e.g. MC7 might be the lsb of the address bits)
- MM1, MM0: two MemoryAccess configuration bits

Depending on the Memory Access mode, up to six blocks of 0.5Mbytes can be accessed = three Mbytes in total; the selection of the appropriate block is done with the control signals MC7..0, performing the function of 'chip select' signals (see also table further on).

The MemoryAccess input pins must be strapped to VDD/VSS in order to select one of four possible access modes. The different modes allow interfaces to simple and common byte-oriented external memories, or more advanced two-byte-oriented memories. Note however that the selected MemoryAccess mode is valid for each block and all external memories : each block in the MTC-20280 address space is handled in the same way.

Basically, the following four modes are supported by the MTC-20280:

Case	MM1	MM0	Description
a:	0	0	WordAccess Disabled (only single-byte access possible)
b:	1	1	WordAccess Enabled with ChipSelect/Low/Up control outputs
c:	1	0	WordAccess Enabled with ChipSelect/Nbyte/Addr0 control outputs
d:	0	1	WordAccess Enabled with ChipSelectUp/ChipSelectLow control outputs

When the WordAccess Mode is disabled, the MTC-20280 Memory interface will always convert the 8-, 16- or 32-bit access of the ARM into one, two or four consecutive SingleByte external accesses.

When the WordAccess Mode is enabled, the MTC-20280 memory interface will decide which type of access is performed. TwoByte (using the full 16bit data bus DQ[15:0]) or SingleByte mode (using only eight bits of the 16bit data bus). One SingleByte access for an 8-bit ARM access, and one or two TwoByte accesses for 16- and 32-bit ARM access. The alignment of the bytes onto the data bus depends on the selected mode, as shown below in the table.

The three types of WordAccess Mode Enabled allow control of different types of external memory by the MTC-20280, each of them connected and controlled in a different way. Therefore, the meaning of the control signals MC7..0 that are sent to the external memory will depend on the selected type.

The following memory types and configurations are supported:

case a: (see Figure 15)
normal byte-oriented external memory (controlled by CS,OE,WE) one memory allocated to store both LSB and MSBytes addressed in SingleByte access mode only (WordAccess Disabled).

case b: (see Figure 16)
normal byte-oriented external memory (controlled by CS,OE,WE) two memories allocated (one for LSByte, one for MSByte) addressed in SingleByte or TwoByte access (WordAccess Enabled) requires glue logic for chip-select signal generation in between the MTC-20280 and the memory.

case c: (see Figure 17) external memory with integrated single / double byte access mode (controlled by CS,OE,WE,BYTE,A0) (e.g. AMD or INTEL memories) addressed in SingleByte or TwoByte access (WordAccess Enabled).

case d: (see Figure 18) normal byte-oriented external memory (controlled by CS,OE,WE) with two memories allocated (one for LSByte, one for MSByte) addressed in SingleByte or TwoByte access (WordAccess Enabled) no glue logic for chip-select signal generation in between the MTC-20280 and the memory needed.

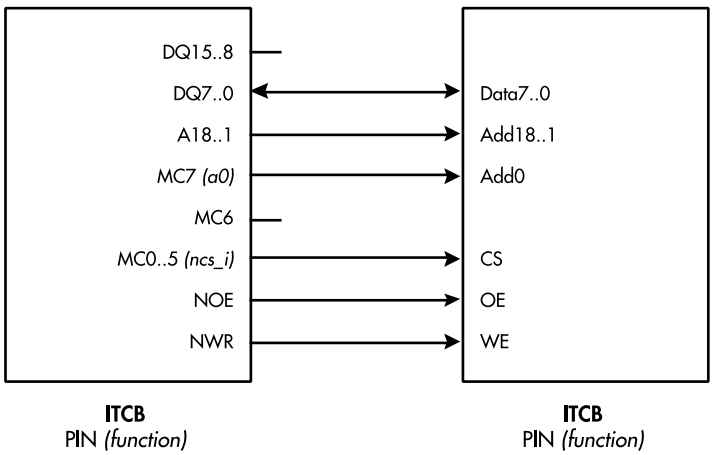


Figure 15 : MTC-20280 - External Memory Connections for 'Case a'.

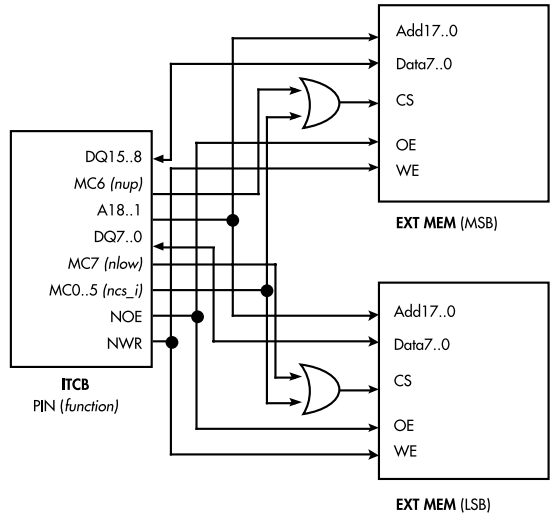


Figure 16 : MTC-20280 - External Memory Connections for 'Case b'.

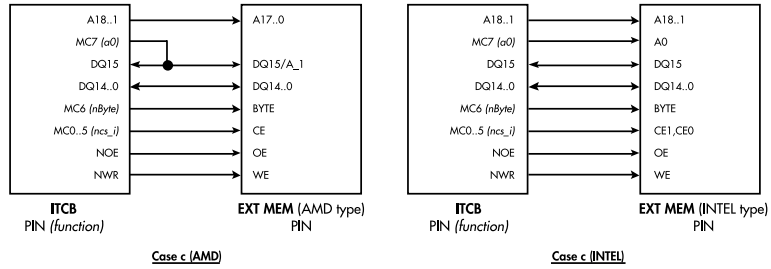


Figure 17 : MTC-20280 - External Memory Connections for 'Case c'.

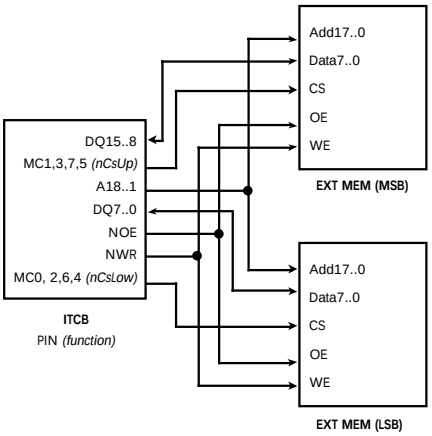


Figure 18 : MTC-20280 - External Memory Connections for 'Case d'.

The output pins have the following logical meaning dependent on the mode:

	MM		NWR	NOE	MC0	MC1	MC2	MC3	MC4	MC5	MC6	MC7	SingleByte		TwoByte		
													A	DQ	DQ	DQ	DQ
	1:	0											18:1	15:8	7:0	15:8	7:0
a	0	0	nwr	noe	ncs0	ncs1	ncs2	ncs3	ncs4	ncs5	x	a0	addr 18:1	pio	b_i	--	--
b	1	1	nwr	noe	ncs0	ncs1	ncs2	ncs3	ncs4	ncs5	nup	nlow	addr 18:1	b_1 b_3	b_0 b_2	b_1 b_3	b_0 b_2
c	1	0	nwr	noe	ncs0	ncs1	ncs2	ncs3	ncs4	ncs5	nbyte	a0*	addr 18:1	z	b_i	b_1 b_3	b_0 b_2
d 0	1		nwr	noe	ncl0	ncu0	ncl1	ncu1	ncl3	ncu3	ncl2	ncu2	addr 18:1	b_1 b_3	b_0 b_2	b_1 b_3	b_0 b_2

Meaning of the codes:

SingleByte selected by the memory interface controller:
alignment of any byte_i to be transferred to/from external memory at a given time
TwoByte selected by the memory interface controller:
alignment of any byte_i to be transferred to/from external memory at a given time

-- : non-occurring situation
x : means the output is not used in this mode; it will not be floating but set to VDD
z : means the bidirectional IO is not used in this mode; it will be set to tri-state (input)
pio : pins used as parallel IO: ports DQ15..12=PE3..0 and DQ11..8=PF3..0
a0 : LSB of address
a0* : the LSB of the address will not be used and be put in tri-state, if and only if a TwoByte access is selected; this is done to be compatible with AMD memories, for which DQ15 and a0 must be connected to the same pin (ref [13])

ncs<i>: active low chip select for block <i> in the memory map (both LSByte and MSByte)
ncl<i> : active low chip select for block <i> in the memory map (only LSByte)
ncu<i> : active low chip select for block <i> in the memory map (only MSByte)
nlow : active low indication that lower byte is transferred
nup : active low indication that upper byte is transferred
nwr : active low indication for write enable
noe : active low indication for read enable
nbyte : active low indication for SingleByte transfer selection, TwoByte (nbyte=1) or SingleByte (nbyte=0)

The following logical relationship must hold between the signals:

nlow = (nbyte + a0)*not(nbyte);
nup = (nbyte + not(a0))*not(nbyte);
ncl<i> = (ncs<i> + nlow);
ncu<i> = (ncs<i> + nup);

Note that in case d, each chip select output must be split into two signals: one for the lower byte and one for the upper byte. Re-use of pins MC6 and MC7 allows the user to split the select signals for the lower four blocks of the memory map; blocks 5 and 6 of the external memory map will not be addressable. Instead, one obtains the advantage that no glue logic is needed to control the chip select pins of the external memories.

Wait state(s) per memory block:
For each of the six external blocks, wait cycles can be specified in order to optimize the external components configuration. The registers CHIP_WTC1, 2 and 3 allow up to 15 wait cycles to be inserted per block; the default value after reset is 15 wait cycles for each block.

Memory access timing.
The drawing below shows the timing relationship of the memory interface for a 2-byte access, making use of the Nbyte control signal:

- Timing compatible with standard SRAM and Flash, NCS controlled.
- When no wait state is specified, the NCS strobe correspond to the low level of the internal ARM clock; the width of the low and the high level of NCS is symetric and correspond to the selected ARM clock speed.
- When wait states are specified, an equivalent number of ARM cycles is added to the external access timing; so ONLY the low level of NCS is extended (as required for slower SRAMs). It's always possible to keep the levels symetric if neede by instead of placing wait states, decreasing the ARM clock frequency.

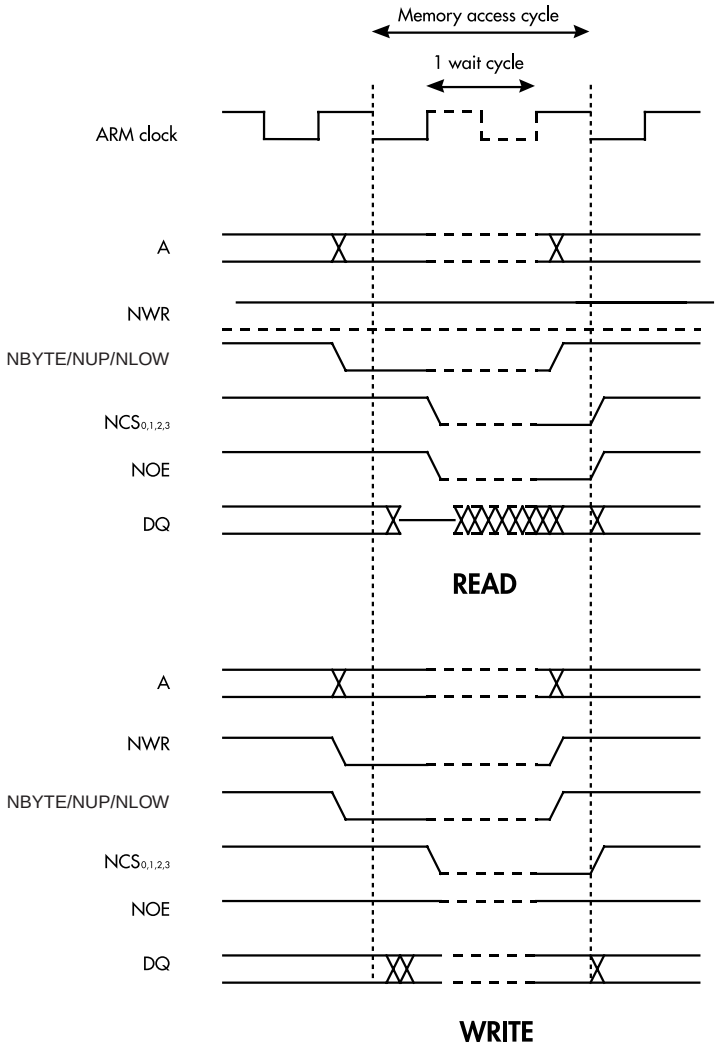


Figure 19 : Memory Interface Signals – Read and Write Cycles

Memory space organisation
The most significant bits of the 32-bit internal ARM address word, are decoded to split the memory space into 8 blocks assigned as follows:

0: external block 0
1: external block 1
2: external block 2
3: external block 3
4: external block 4
5: external block 5
6: internal fast memory
7: APB bus

When the ARM accesses one of the external blocks <i>, the MTC-20280 will bring the corresponding chip select control signal nCS<i> active low. Note that there are no gaps between addresses of successive external memory blocks. For both the 'internal fast memory' and the 'APB memory' a full length page of 0x1000 0000 bytes is allocated, although the current MTC-20280 design uses only 0x400 and 0x100 bytes respectively for these memory banks. This is done to ensure backward compatibility in future versions of the MTC-2028x controller family.

Data in the 'internal RAM' can be either a byte, half-word or word; their addresses are LSB aligned, making use of the 10 LSB of the ARM address bus (from 0xC000 0000 up to 0xC000 03FF). All data of the 'APB memory' are bytes; their addresses are 'word-aligned' (LSB+2bits) such that the data byte is always LSB aligned onto the data bus. The ARM address bits [9..2] are used to address the full memory map, the 2 LSB of the address being zero (from 0xE000 0000 up to 0xE0000 03FC). At reset the ARM will start-up at the fixed boot address 0x0000 0000, accessing the external block 0.

In order to allow re-definition of the code in external block 0 (e.g. for reprogramming of the interrupt vectors), the addresses of block 0 can be swapped with block 3 under software control. This is done as soon as the bit CHIP_CFG[7]=1 is set. At reset, the blocks are not swapped.

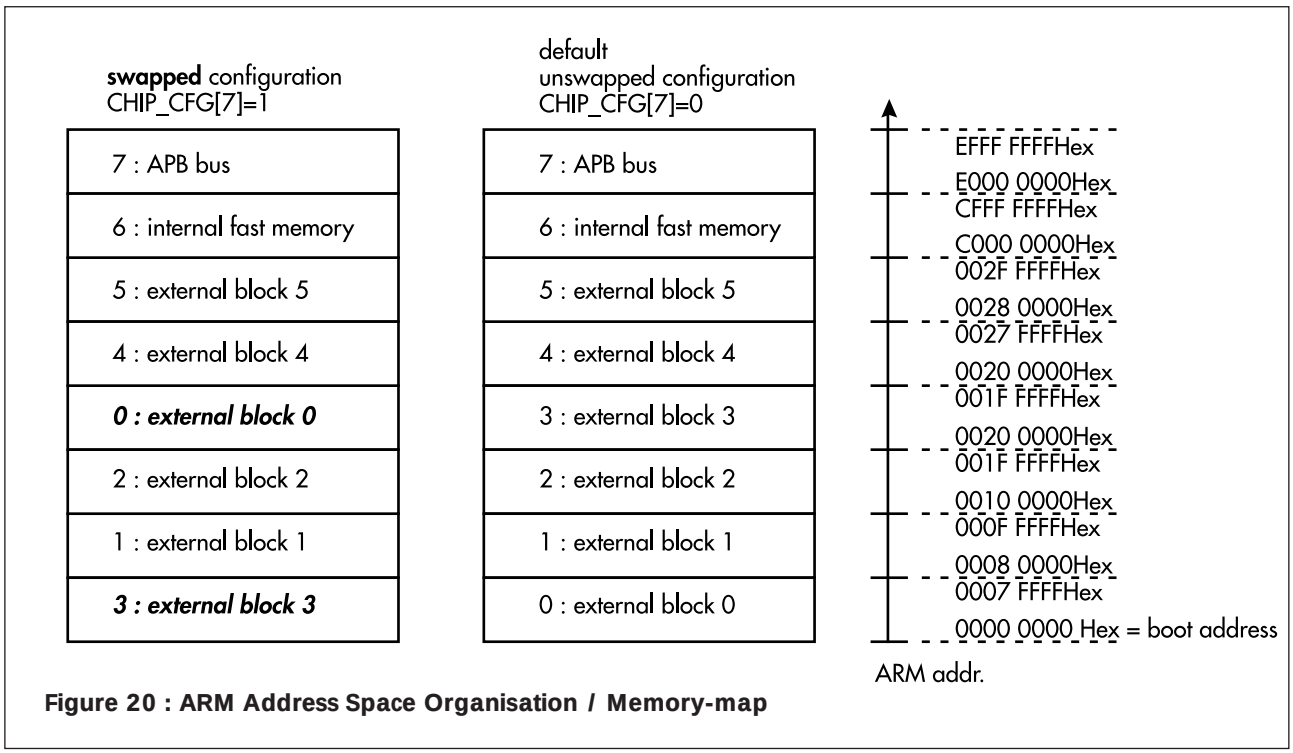


Figure 20 : ARM Address Space Organisation / Memory-map

Register table:

Name	Address		Function
	word	byte	
CHIP_CFG[7]	01	4	Swap addresses for external memory block 0 and 3 (1 bit): 0=unswapped (default at reset, address 00 in block 0), 1=swapped (address 00 in block 3)
CHIP_WTC1	04	10	Wait cycles for external memory blocks (default 15: slow at reset) bit[3:0]: for memory block 0 bit[7:4]: for memory block 1
CHIP_WTC2	05	14	Wait cycles for external memory blocks (default 15: slow at reset) bit[3:0]: for memory block 2 bit[7:4]: for memory block 3
CHIP_WTC3	06	18	Wait cycles for external memory blocks (default 15: slow at reset) bit[3:0]: for memory block 4 bit[7:4]: for memory block 5

Interrupts

The chip contains several interrupt sources. These sources are split into two types of interrupts: a fast (NFIQ) and normal (NIRQ) interrupt. Priority within one class is resolved in software. Interrupts are controlled by writing to / reading from a set of control/status registers: IRQ1_* for NFIQ, and IRQ2_* for NIRQ.

Each source can be masked by clearing a bit in a control register (ENCLR), or unmasked by setting a bit (ENSET). At reset, all sources are masked by default. All 'interrupt request' bits from the various sources are readable in a register, and each of them can be cleared. Both the masked (ST) and the unmasked 'raw' (STR) interrupt status can be checked. As soon as a non-masked interrupt occurs, the NFIQ / NIRQ pin of the ARM goes active low. The interrupt can be cleared by writing to the acknowledge control register (ACK). If all interrupts in irqStatus are cleared, the NFIQ / NIRQ pin goes inactive high again.

Register table:

Name	Address		Function
	word	byte	
IRQ1_ST	94 (R only)	250	Interrupt status for sources after masking with IRQ1_EN (IRQ1_ST(i)=1 if interrupt(i) is active after masking)
IRQ1_STR	95 (R only)	254	Interrupt status for sources without masking (IRQ1_STR(i)=1 if interrupt(i) is active)
IRQ1_EN	96 (R only)	258	Interrupt mask register used to generate IRQ1_ST (IRQ1_EN(i)=1 if interrupt(i) must not be masked)
IRQ1_ENSET	97 (W only)	25C	Control register to unmask an interrupt (IRQ1_ENSET(i)=1 results in IRQ1_EN(i)=1, i.e. unmask interrupt(i))
IRQ1_ENCLR	98 (W only)	260	Control register to mask an interrupt (IRQ1_ENCLR(i)=1 results in IRQ1_EN(i)=0, i.e. mask interrupt(i))
IRQ1_ACK	99 (W only)	264	Control register to clear an interrupt (IRQ1_ACK(i)=1 resets the interrupt(i) detection in irq(Raw)Status(i))
IRQ2_ST	9A (R only)	268	Interrupt status for sources after masking with IRQ2_EN (IRQ2_ST(i)=1 if interrupt(i) is active after masking)
IRQ2_STR	9B (R only)	26C	Interrupt status for sources without masking (IRQ2_STR(i)=1 if interrupt(i) is active)
IRQ2_EN	9C (R only)	270	Interrupt mask register used to generate IRQ2_ST (IRQ2_EN(i)=1 if interrupt(i) must not be masked)

Name	Address		Function
	word	byte	
IRQ2_ENSET	9D (W only)	274	Control register to unmask an interrupt (IRQ2_ENSET(i)=1 results in IRQ2_EN(i)=1, i.e. unmask interrupt(i))
IRQ2_ENCLR	9E (W only)	278	Control register to mask an interrupt (IRQ2_ENCLR(i)=1 results in IRQ2_EN(i)=0, i.e. mask interrupt(i))
IRQ2_ACK	9F (W only)	27C	Control register to clear an interrupt (IRQ2_ACK(i)=1 resets the interrupt(i) detection in irq(Raw)Status(i))

Interrupt mapping

NFIQ Interrupt sources (bit k=0..7 of registers IRQ1_*[k] with k=0=LSB)

0 : GCI interrupt generated each time a new GCI frame starts

1 : HDLC1 formatter interrupt generated by HDLC1 formatter (see module description for the various sources of interrupts). The HDLC interrupt registers can be read to determine which interrupt has occurred.

2 : HDLC2 formatter idem as for HDLC1

3 : HDLC3 formatter idem as for HDLC1

4 : Timer1 Timer 1 reaches the limit

5 : PA Parallel IO A interrupt is generated as soon as one of the input bits is toggled (b)

6 : PF Parallel IO F [only applicable if MM[1,0]= 00] interrupt is generated as soon as one of the input bits is toggled

7 : NA

NIRQ Interrupt sources (bit k=0..7 of registers IRQ2_*[k] with k=0=LSB)

0 : UART interrupt generated by UART (see module description to know various sources of interrupts). The UART interrupt register can be read to determine which interrupt has occurred.

1 : PB Parallel IO B interrupt is generated as soon as one of the input bits is toggled

2 : External Rising / falling edge or IRQ high/low level detection on the MTC-20280 pin IT

3 : C/I Activity detected on one activity of the C/I channels not detection masked. The GCI interrupt register can be read to determine which channel is concerned.

4 : Timer2 Timer 2 reaches the limit

5 : PC Parallel IO C. Interrupt is generated as soon as one of the input bits is toggled

6 : PD Parallel IO D. Interrupt is generated as soon as one of the input bits is toggled

7 : PE Parallel IO E [only applicable if MM[1,0]= 00]. Interrupt is generated as soon as one of the input bits is toggled

External interrupt

The external interrupt can be made edge or level sensitive by writing the appropriate value to the following control register :

Register table

Name	Address		Function
	word	byte	
CHIP_CFG	01	4	Type of external interrupt to detect (2bit):

UART Interface

The chip contains a UART (Universal Asynchronous Receiver / Transmitter) compatible with the popular "16550" family, which is fully programmable by the ARM CPU.

Features

Word lengths from 5 to 8 bits
Parity bit : even, odd or forced to a defined state
One or two stop bits
Programmable Baud rate generator (19.2 kbps, 57.6 kbps, 115.2 kbps, 230.4 kbps, ...)
Modem control signals RTS and CTS available
Two, 16 bytes FIFOs, one for transmit and one for receive
Interrupt generated from any one of 10 sources of the UART module itself

Baud rate generator

The baud rate is specified by the divisor register DL (DLM = MSB; DLL = LSB, see Register table below) in the following way:

baud rate = (230400 / DL) bps

Baud Rate (bps)	Division Factor (DL)
2400	96
4800	48
9600	24
19200	12
38400	6
57600	4
115200	2

Register table

Name	Address		Function
	word	byte	
UART_RBR	50 (R only) [if DLAB=0]	140	Receiver Buffer Register, This register is updated from the receiver shift register at the end of a receive sequence.
UART_THR	50 (W only) [if DLAB=0]	140	Transmitter Holding Register, Data is held in this register until transferred to the transmitter shift register
UART_IER	51	144	Interrupt Enable Register [if DLAB=0] = [x, x, x, x, EDSSI, ELSI, ETBEI, ERBFI] EDSSI : Enable Modem Status Interrupt When set ('1'), an UART interrupt is generated if D0, D1, D2, or D3 of the Modem Status Register become set. ELSI : Enable Rx Status Interrupt When set ('1'), an interrupt is generated if D1, D2, D3 or D4 of the Line Status Register become set. ETBEI : Enable Tx Holding Register Empty Interrupt When set ('1'), an interrupt is generated if THRE=1 or the Transmitting Holding Register is empty. ERBFI Enable Receiver Buffer Register When set ('1'), an interrupt is generated if the Receive Buffer contains data.

Name	Address		Function
	word	byte	
UART_IIR	52 (R only)	148	Interrupt Identification Register = [FIFOE, FIFOE, 0, 0, ID2, ID1, ID0, NINT] FIFOE Returns 1 if FIFOs enable, otherwise 0 ID[2:0] : Interrupt ID 0 : Modem status. Interrupt cleared when reading the modem status register 1 : Transmitter holding register empty. Interrupt cleared when reading this register or when writing to the transmitter holding register 2 : Receive Data available or Rx FIFO trigger. Interrupt cleared when reading receive buffer register 3 : Receiver line status. Interrupt cleared when reading the line status register 4 : / 5 : / 6 : Character timeout indication. Interrupt cleared when reading receive buffer register 7 : / NINT Interrupt pending, active low Notes: Receive timeout interrupt occurs if all the following apply: - there is at least one character in the FIFO - the most recent character was received longer than 4 character periods ago (inclusive of all start, parity, and stop bits) - the most recent CPU read of the FIFO was longer than 4 character periods ago The timeout timer is restarted on receipt of a new byte from the input shift register, or on a CPU read from the Rx FIFO. TX FIFO interrupt occurs when Tx FIFO is empty. This interrupt will be delayed one character period minus the last stop bit period whenever; THRE=1 and there have not been at least 2 bytes in the Tx FIFO at the same time since the last time THRE=1. If the Tx interrupt is enabled, setting bit 0 of the FCR will generate an immediate interrupt. If the FIFOs are enabled and at least one of the active bits in IER is disabled, then the UART will operate in the FIFO polled mode. Since the Tx and Rx paths are controlled separately, either one or both can be in the polled mode. The application software should check Tx and Rx status using the LSR.
UART_FCR	52 (W only)	148	FIFO Control Register = [RFTL1, RFTL0, x,x, DMA1, CLRT, CLRR, FIFOE] RFTL[1:0] : RX FIFO trigger level Defines RX FIFO trigger level in number of bytes 00 : 01 bytes 01 : 04 bytes 10 : 08 bytes 11 : 14 bytes DMA1 Set DMA mode 1. In MTC-20280 configuration, no DMA support is provided. CLRT : Clear TX FIFO CLRR : Clear RX FIFO

Name	Address		Function
	word	byte	
			FIFOE Enable FIFOs; when the FIFOs are enabled or disabled, both Rx and Tx FIFOs are reset. This bit must be a 1 for any of the other bits in the register to have any effect.
UART_LCR	53	14C	Line Control Register = [DLAB, SB, SP, EPS, PEN, STB, WLS1, WLS0] DLAB : Divisor Latch Access Bit When clear '0', Receive and Transmitter Registers are read / written address 50 and IER register at address 51. When set '1', Divisor Latch LS is read / written at address 50 and Divisor Latch MS read / written at address 51. SB : Set Break When set '1', TXD signal is forced into the '0' state SP : Stick Parity When set '1', parity bit is forced into a defined state, dependent upon state of EPS, PEN : If EPS='1' & PEN='1' parity bit is set and checked = '0' If EPS='0' & PEN='1' parity bit is set and checked = '1' EPS : Even Parity Select When set '1' and PEN = '1' an even number of ones is sent and checked. When clear '0' and PEN = '1' an odd number of ones is sent and checked. PEN : Parity Enable When set to '1', parity is transmitted and checked. Parity bit is added after the data field and before the STOP bits. When clear '0' parity is neither transmitted nor checked. STB : Number of STOP bits When set '1' two STOP bits are added after each character is sent, except if character length is 5, then 1_ STOP bits are added. When clear '0' one STOP bit is always added. Only the transmit STOP bits are programmable, the receiver stage only expects one STOP bit irrespective of the state of STB. WLS[1:0] : Word Length Select Transmitted and received character size defined as follow: 00 = 5 bit 01 = 6 bit 10 = 7 bit 11 = 8 bit
UART_MCR	54	150	Modem Control Register = [X, X, X, LOOP, OUT2, OUT1, RTS, DTR] LOOP : Loop back mode When set '1' the following conditions are implemented: TXD is forced to '1' RXD is disconnected from the Rx input shift register the Rx input shift register is connected to the Tx output shift register the modem status signals are disconnected the modem control signals are connected to modem status inputs OUT[2:1] Not used RTS Control the state of the corresponding output, even in loop mode. DTR Control the state of the corresponding output, even in loop mode.

Name	Address		Function
	word	byte	
UART_LSR	55	154	Line Status Register = [FIFOERR, TEMT, THRE, BI, FE, PE, OE, DR] FIFOERR : RX Data Error in FIFO This bit is set to 1 when there is at least one PE, FE, or BI in the RX FIFO. It is cleared by a read from the LSR register, if there are no subsequent errors in the FIFO. TEMT : Transmitter Empty If the FIFOs are disabled, this bit is set to '1' whenever the transmitter holding register and the transmitter shift register are empty. If the FIFOs are enabled, this bit is set whenever the TX FIFO and the transmitter shift register are empty. In both cases, this bit is cleared when a byte is written to the TX data channel. THRE : Transmitter Holding Register Empty If the FIFOs are disabled, this bit is set to '1' whenever the transmitter holding register is empty and ready to accept new data, this bit is cleared when the data is transferred to the transmitter shift register. If the FIFOs are enabled, this bit is set to '1' whenever the TX FIFO is empty. It is cleared when at least one byte is written to the TX FIFO. BI : Break Interrupt If the FIFOs are disabled, this bit is set whenever the RXD is held in the 0 state for more than a transmission time (START bit + DATA bits + PARITY + STOP bits). BI is reset by the CPU reading this register. If the FIFOs are enabled, this error is associated with the corresponding character in the FIFO. The error is flagged when this byte is at the top of the FIFO. When break occurs, only one zero character is loaded into the FIFO. The next character transfer is enabled when RXD goes into the marking state and receives the next valid start bit. FE : Framing Error If the FIFOs are disabled, this bit is set if the received data did not have a valid STOP bit, FE is reset by the CPU reading this register. If the FIFOs are enabled, the state of this bit is revealed when the byte it refers to is at the top of the FIFO. PE : Parity Error If the FIFOs are disabled, this bit is set if the received data does not have a valid parity bit, PE is reset by the CPU reading this register. If the FIFOs are enabled, the state of this bit is revealed when the byte it refers to is at the top of the FIFO OE : Overrun Error If the FIFOs are disabled, this bit is set if the receive buffer was not read by the CPU before new data from the receiver shift register overwrote previous contents. OE is cleared when the CPU reads this register. If the FIFOs are enabled, an overrun error occurs when the RX FIFO is full and the RX shift register becomes full. OE is set as soon as this happens. The character in the shift register is then overwritten, but is not transferred to the FIFO. DR : Data Ready This bit is set whenever the receive buffer is full, or by a byte being transferred into the FIFO. DR is cleared by the CPU reading the receive buffer or by reading all of the FIFO bytes. This bit is also cleared whenever the FIFO enable bit is changed.

Name	Address		Function
	word	byte	
UART_MSR	56	158	Modem Status Register = [DCD, RI, DSR, CTS, DDCD, TERI, DDSR, DCTS] DCD : Data Carry Detect When Loop = '0' this is the input signal DCD When Loop = '1' this is equal to OUT2 (not used) RI When Loop = '0' this is the input signal RI When Loop = '1' this is equal to OUT1 (not used) DSR : Data Set Ready When Loop = '0' this is the input signal DSR When Loop = '1' this is equal to DTR CTS : Clear To Send When Loop = '0' this is the input signal CTS When Loop = '1' this is equal to RTS DDCD : Delta Data Carry Detect This bit is set ('1') if the state of DSR has changed since this register was last read. TERI : Trailing Edge Ring Indicator This bit is set if the RI input changes from '1' to '0' since this register was last read. DDSR : Delta Data Set Ready This bit is set ('1') if the state of DSR has changed since this register was last read. DCTS : Delta Clear to Send This bit is set ('1') if the state of CTS has changed since this register was last read.
UART_SCR	57	15C	Scratch Register General-purpose read/write register, undefined after reset.
UART_DLM UART_DLL	51 50 (if DLAB=1)	144 140	Divisor Latch , MSB and LSB for baud-rate control (see table) After reset DLM,DLL are undefined.

Parallel I/O ports

Four times 4-bit bidirectional I/O ports are provided (PA[3..0], PB[3..0], PC[3..0], PD[3..0]) to allow additional, user-defined functions. The application software can define the access direction of any of the four pins and have a total visibility of the pin activity (read and write access). One interrupt signal is generated per parallel I/O source.

In case the WordAccess of the external memory interface is disabled (case a), the MSB of the data bus is used for 2 more 4-bit parallel I/O ports (PE[3..0], PF[3..0]) ; these ports have the same functionality as the other parallel I/O ports. Whenever the WordAccess mode of the memory interface is enabled, the registers controlling the ports PE and PF become irrelevant and no interrupt signal will be generated.

The port PB has an extra function: by setting the bit CHIP_CFG[0]=1, this port can also be used for UART extended modem control signals.

Pin	Direction	UART-pin and function (all active high and non inverted)
PB0	Input	DCD: Data carrier detect input of UART
PB1	Input	RI: Ring indicator input of UART
PB2	Input	DSR: Data set ready input of UART
PB3	Output	DTR: Data Terminal Ready output of UART

In this case, the value of PB_DIR is by-passed. As soon as CHIP_CFG[0] becomes zero again, the actual value of PB_DIR will be used as before. At reset, CHIP_CFG[0]=0, which corresponds to the non-UART mode.
Similarly, the port PA has an extra function related to the Timers: by setting any of the four configuration bits CHIP_CFG[6..3]=1, the corresponding PA bit is connected to either a Timer input or output.

Pin	Direction	Timer function
PA0	Output	Connected to T2O if CHIP_CFG[3]=1
PA1	Input	Connected to T2I if CHIP_CFG[4]=1
PA2	Output	Connected to T1O if CHIP_CFG[5]=1
PA3	Input	Connected to T1I if CHIP_CFG[6]=1

The value of PA_DIR[k] is by-passed at that moment CHIP_CFG[k+3]=1. As soon as CHIP_CFG[k+3] becomes zero again, the actual value of PA_DIR[k] will be used as before. At reset, CHIP_CFG[k+3]=0, which corresponds to the normal non-timer mode.

Register table

Name	Address		Function
	word	byte	
PAB_OUT	A0 [3..0]	280	Data register used to drive PB[3..0] set in output direction
PAB_IN	A1 [3..0]	284	Data register to store value of PB[3..0]
PAB_DIR	A2 [3..0]	288	Selection of the direction of each bit. Biti = 0 : Pbi set to input node 1 : Pbi set to output node reset value = 0 hex : all bits set in INPUT mode
CHIP_CFG[0]	01	4	bit[0] : Control bit to connect PB with UART (=1) or not (=0)

Name	Address		Function
PAB_OUT	A0 [7..4]	280	Data register used to drive PA[3..0] set in output direction
PAB_IN	A1 [7..4]	284	Data register to store value of PA[3..0]
PAB_DIR	A2 [7..4]	288	Selection of the direction of each bit. Biti = 0 : Pai set to input node 1 : PAi set to output node reset value = 0 hex : all bits set in INPUT mode
CHIP_CFG[6..3	01	4	bit[k+3] : Control bit to connect PA[k] with a Timer (=1) or not (=0)

Name	Address		Function
PCD_OUT	A3 [7..4]	282	Data register used to drive PC[3..0] set in output direction
PCD_IN	A4 [7..4]	290	Data register to store value of PC[3..0]
PCD_DIR	A5 [7..4]	294	Selection of the direction of each bit. Biti = 0 :PCi set to input node 1 :PCi set to output node reset value = 0 hex : all bits set in INPUT mode
PCD_OUT	A3 [3..0]	282	Data register used to drive PD[3..0] set in output direction
PCD_IN	A4 [3..0]	290	Data register to store value of PD[3..0]
PCD_DIR	A5 [3..0]	294	Selection of the direction of each bit. Biti = 0 : PDi set to input node 1 : PDi set to output node reset value = 0 hex : all bits set in INPUT mode

Name	Address		Function
	word	byte	
PEF_OUT	A6 [7..4]	298	Data register used to drive PE[3..0] set in output direction
PEF_IN	A7 [7..4]	29C	Data register to store value of PE[3..0]
PEF_DIR	A8 [7..4]	2A0	Selection of the direction of each bit. Biti = 0 :PEi set to input node 1 :PEi set to output node reset value = 0 hex : all bits set in INPUT mode
PEF_OUT	A6 [3..0]	298	Data register used to drive PF[3..0] set in output direction
PEF_IN	A7 [3..0]	29C	Data register to store value of PF[3..0]
PEF_DIR	A8 [3..0]	2A0	Selection of the direction of each bit. Biti = 0 : PFi set to input node 1 : PFi set to output node reset value = 0 hex : all bits set in INPUT mode

Timers

Timers 1 and 2

Down-counting timers with interrupt generation, event-count and –capture functions

Two timers are provided (Timer1 and Timer2). Both are 16-bit load-able down-counters which count down as long as the counter is enabled (count=1). They restart automatically by presetting to the initial_value when timer_value == 00 is reached. They generate an interrupt request on timeout (whenever timer_value == 00). They also generate a 'toggling' output signal (T1O, T2O), that changes state at each timeout occurrence. The output signal can be used in the external application via the Parallel I/O port A. Timers can be read on the fly (no need to put count=0 to read). By default, the counters operate on "internal count mode", where counting is based on one of two possible fixed counting frequencies :

slow internal mode (fast=0): 960 kHz (=15.36 MHz / 16), default after reset
fast internal mode (fast=1): 3840 kHz (=15.36 MHz / 4)

In slow internal mode, the period between timeout / toggling events can be up to 68 ms (216 x (960 kHz)-1, or about 1 µs per count step) .

The timers can be set to count rising edges of an input signal (T1I, T2I) which is applied on the parallel I/O port A. The "external count mode" is selected as soon as the CHIP_CFG bit corresponding to T1I or T2I is set. The input signal is sampled at 3840 kHz for the detection of a rising edge. Two variants exist:

slow external mode (fast=0): the counter is decreased every fourth time a rising edge is detected
fast external mode (fast=1): the counter is decreased every single time a rising edge is detected

In this mode the counters may be reset or stopped "on-the-fly".

Furthermore, Timer1 has an extra capture register which can be used to store the actual value of the timer register when the capture control signal (capSig, generated by a hardware event) becomes true. Capturing is enabled by the control bit capEn (capEn=1). The following sources can be used to generate the capSig signal:

- the CI activity detection signal (detection of change in CI bit values
- the Parallel I/O A detection signal (detection of change on bits, set in input mode)
- the External interrupt input signal (either rising/falling edge or high/low level detection)
- the Parallel IO B detection signal (detection of change on bits, set in input mode)

These four signals correspond to the unmasked, 'raw' interrupt status bits.

Capturing is done only once, for the first event occurring since the capEn bit was set by software. Once captured, the bit is automatically reset by the hardware. This prevents the timer from capturing twice before reading / re-enabling the capturing registers. Moreover, the bit also has an acknowledge function: reading the bit value permits a check on whether a capture event occurred or not.

After reset, the counters are initialised as follows:

- Timer CMD register (TI_CMD) = 0x00 (no-count, not-init, capturing disabled, slow (internal) mode)
- Internal mode selected because at reset CHIP_CFG[6,4]=0
- Timer registers (TI_1L, TI_1M, TI_2L, TI_2M) = 0xFF
- Timer1 capture registers (TI_C1L, TI_C1M) = 0xFF

Watch-dog timer
A watchdog timer is provided, working on the ApbClock (with frequency = Xtal/4 or 3.84 Mhz = 0.26 ns). It is a countdown timer, which is by default disabled after (power-up) reset of the MTC-20280.

It can be disabled, enabled and 'kicked' under software control by writing a specific "magic" word value to the WD_MAGIC register. This changes the state of the watchdog finite state-machine (FSM), of which the state can be monitored by reading the state bit values in the WD_SC status/control register.

Disabling is achieved by writing a sequence of three words to the WD_MAGIC register ("magic1-word", "magic2-word", and "magic3-word"). This is to avoid accidental disabling of the timer. If the sequence is interrupted by writing any other value to the WD_MAGIC register, it must be restarted in order to disable the watchdog successfully. However, the sequence may be interleaved without harm by write commands to other registers, or by read commands to any register including WD_MAGIC itself.

Enabling is done by a single write of any value to the WD_MAGIC register.

Once enabled, the watchdog can be 'kicked' by writing the specific 'kick-value' to WD_MAGIC. 'Kicking' the watchdog performs a re-initialisation at one of four possible values: This prevents the timer from reaching the zero value, which will generate a reset for the ARM. The initial value is selected by the two control bits in the WD_SC register.

The output of the watchdog timer controls the active-low NRESET pin of the ARM processor and resets all functional MTC-20280 blocks. It pulses active low, as soon as the timer reaches the zero value. It has the same function as applying an external hardware reset on the pin NRST and is externally available for system reset.

Caution: As the NRST pin can be pulled LOW by a watchdog timeout, the application circuit should contain a resistor in series with the pin when an external capacitor is used to form a power-on reset circuit. (The NRST pin will attempt to discharge this capacitor very quickly causing a high peak current, which may result in damage to the pin driver. The resistor limits the current to safe values).

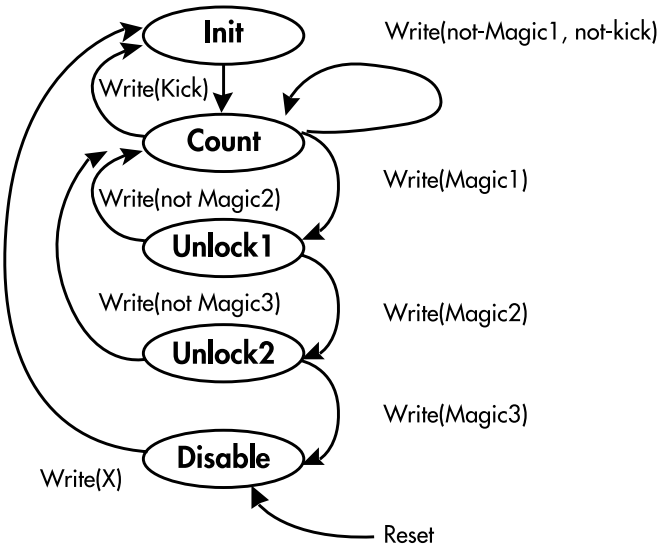


Figure 21 : Watchdog Finite State Machine

Register table

Name	Address		Function
	word	byte	
TI_1L TI_1M	C0 C1	300 304	Timer 1 counter 16 bits (TI_1: LSB & MSB) Read: counter value Write: initial value
TI_2L TI_2M	C2 C3	308 30C	Timer 2 counter 16 bits (TI_2: LSB & MSB) Read: counter value Write: initial value
TI_CMD	C4	310	Timers command register: Timer 1: bit[0]: count (=1:counting; =0:not counting) bit[1]: init (set to 1 to re-initialise; self-clearing bit) bit[2]: fast (=1: fast clock = Xtal / 4; =0: slow clock = Xtal / 16) bit[3]: capEn (=1:capture enabled; =0:disabled) (cleared by HW when captured) Timer 2: bit[4]: count (=1:counting; =0:not counting) bit[5]: init (set to 1 to re-initialise; self-clearing bit) bit[6]: fast (=1: fast clock = Xtal / 4; =0: slow clock = Xtal / 16) bit[7]: / (not used)
TI_C1L TI_C1M	C5 C6	314 318	Timer 1 capture registers: 16 bits (LSB & MSB) Read only
TI_CSEL	C7	31C	capture source selection register : 00 : CI change (default) 01 : external interrupt IT 10 : PB input port change 11 : PA input port change
WD_SC [3.. 0]	C8	320	Watch-dog Status and Control register (4 bit) bit[1.. 0] : Init_value selection (Write and Read possible) 00 : 0.48 sec 01 : 0.61 sec 10 : 0.89 sec 11 : 1.02 sec (default) bit[3.. 2] : State of watch-dog FSM (Read only) 00 : init & count 01 : unlock1 10 : unlock2 11 : disable (default)

WD_MAGIC	C9	324	Watch-dog Magic word register (8bit);
			gives a command to the watch-dog by writing a specific value:
			value=DC : Kick command;
			value=A5 : Magic1-word command;
			value=5A : Magic2-word command;
			value=AF : Magic3-word command;

Hardware identification code
The MTC-20280 contains a read-only register with a hardware identification code. Writing to this register will cause no harm.

The ID code corresponds to the next eight bits:

FFF RR SSS with
FFF = product family code
(001 for MTC-20280)
RR = revision code
(rev.N=00, rev.O=01,
rev.P=10, rev.Q=11)
SSS = mask set version
(000: first, 001: second, ...)

The following ID codes are foreseen for MTC-20280 variants:

MTC-20280 version
HW identification code
MTC-20280.NAA
0x 001 00 000 = 0x20

Register table

Name	Address		Function
	word	byte	
CHIP_ID	00	0	Returns the chip identification code 0x<MN> (Hardware version dependent)

Programming Notes

For information about programming of the ARM7TDMI processor core, please refer to the ARM7 programming manual.

Register map summary

Below is a summary of the complete memory map of the registers discussed in the previous sections.
All parameters that are related to the same hardware unit are grouped together in registers that are in the same 'page'. For details, please see the Detailed Functional Description above.

Addresses not listed in the table should not be used, as they are allocated for possible future upgrades of the memory map for new products.

Registers indicated by a "•" in the first column have a different physical address than the functional equivalent (where appropriate) in the MTC-20270 device.

Register physical addresses

The APB address space is limited to 8 bits wide, corresponding to the ARM address bits 9 to 2 (32-bit word aligned addresses), located in the 7th block of the ARM address space. The data are of type "byte" and are LSB aligned onto the (32-bit) data bus. All other bits (bits 8 to 31) are of undefined state during a read. A write to these bits has no effect. The physical address of an APB register is thus given by ((APB Address)*4 + 0xE0000000). Standard header-files for 'C' or assembler programs that define the register mnemonics and physical addresses are available from Alcatel Microelectronics.

APB address:

bit[3.. 0]: registers selection
bit[7.. 4]: page selection

0: chip configuration (00-06)

1: HDLC1
2: HDLC2
3: HDLC3
4: HDLC extra (40-46)
5: UART (50-57)
6,7,8: GCI router
9: clock gen, interrupt
A: parallel IO's
B: GCI router (ext') (B0-B7)
C: Timers and Watchdog (C0-C9)

	Name	Addr. word	Addr. byte	Access	Width (bits)	Reset (hex)	Function
MTC-20280 Configuration							
•	CHIP_ID	00	0	R	8	<MN>	Returns the chip identification code = 0x<MN> (HW version dependent)
•	CHIP_CFG	01	4	RW	8	06	bit[0]=1 : Configures PB[3.. 0] in PB2Uart mode bit[2,1] = external interrupt (IT) type 00: falling edge 01: rising edge 10: low-level 11:high-level (default) bit[3]=1 : use PA0 as Timer output T2O bit[4]=1 : use PA1 as Timer input T2I bit[5]=1 : use PA2 as Timer output T1O bit[6]=1 : use PA3 as Timer input T1I bit[7]=1 : swap addresses for external memory block 0 and 3
•	CHIP_GCI_L	02	8	R	8	00	Data rate detected on the master GCI interface (data bits per frame). Lsb bits of data rate
•	CHIP_GCI_M	03	C	R	8	00	Msb bits of data rate
•	CHIP_WTC1	04	10	RW	8	FF	Wait cycles for external memory blocks bit[3:0]: mem block 0 bit[7:4]: mem block 1
•	CHIP_WTC2	05	14	RW	8	FF	bit[3:0]: mem block 2 bit[7:4]: mem block 3
•	CHIP_WTC3	06	18	RW	8	FF	bit[3:0]: mem block 4 bit[7:4]: mem block 5
HDLC Formatters							
	HD1_MODE	10	40	RW	8	00	HDLC mode register
	HD1_EMODE	11	44	RW	8	00	HDLC Extended mode register
	HD1_CMD	12	48	W	8	00	HDLC Command register
	HD1_SSTAT	13	4C	R	8	00	HDLC Serial Status register
	HD1_FSTAT	14	50	R	8	32	Fifo Status register
	HD1_ISTAT	15	54	R	8	00	Interrupt Status Register
	HD1_ISRV	16	58	R	8	00	Interrupt Service Request Register
	HD1_IMASK	17	5C	RW	8	00	Interrupt Mask Register
	HD1_FIFO	18	60	RW	8	00	Rx / Tx Fifo data register
	HD1_RFBC	19	64	R	8	00	Receive Frame Byte Count
	HD1_TA1	1A	68	RW	8	00	Transmit Address 1
	HD1_TA2	1B	6C	RW	8	00	Transmit Address 2
	HD1_RAW1	1C	70	RW	8	00	Receive Address Wildcard 1
	HD1_RAW2	1D	74	RW	8	00	Receive Address Wildcard 2
	HD1_RA1	1E	78	RW	8	00	Receive Address Register 1
	HD1_RA2	1F	7C	RW	8	00	Receive Address Register 2
	HD2_MODE	20	80	RW	8	00	HDLC mode register
	HD2_EMODE	21	84	RW	8	00	HDLC Extended mode register
	HD2_CMD	22	88	W	8	00	HDLC Command register

	Name	Addr.		Access	Width(bits)	Reset(hex)	Function
		word	byte				
	HD2_SSTAT	23	8C	R	8	00	HDLC Serial Status register
	HD2_FSTAT	24	90	R	8	32	Fifo Status register
	HD2_ISTAT	25	94	R	8	00	Interrupt Status Register
	HD2_ISRV	26	98	R	8	00	Interrupt Service Request Register
	HD2_IMASK	27	9C	RW	8	00	Interrupt Mask Register
	HD2_FIFO	28	A0	RW	8	00	Rx / Tx Fifo data register
	HD2_RFBC	29	A4	R	8	00	Receive Frame Byte Count
	HD2_TA1	2A	A8	RW	8	00	Transmit Address 1
	HD2_TA2	2B	AC	RW	8	00	Transmit Address 2
	HD2_RAW1	2C	B0	RW	8	00	Receive Address Wildcard 1
	HD2_RAW2	2D	B4	RW	8	00	Receive Address Wildcard 2
	HD2_RA1	2E	B8	RW	8	00	Receive Address Register 1
	HD2_RA2	2F	BC	RW	8	00	Receive Address Register 2
	HD3_MODE	30	C0	RW	8	00	HDLC mode register
	HD3_EMODE	31	C4	RW	8	00	HDLC Extended mode register
	HD3_CMD	32	C8	W	8	00	HDLC Command register
	HD3_SSTAT	33	CC	R	8	00	HDLC Serial Status register
	HD3_FSTAT	34	D0	R	8	32	Fifo Status register
	HD3_ISTAT	35	D4	R	8	00	Interrupt Status Register
	HD3_ISRV	36	D8	R	8	00	Interrupt Service Request Register
	HD3_IMASK	37	DC	RW	8	00	Interrupt Mask Register
	HD3_FIFO	38	E0	RW	8	00	Rx / Tx Fifo data register
	HD3_RFBC	39	E4	R	8	00	Receive Frame Byte Count
	HD3_TA1	3A	E8	RW	8	00	Transmit Address 1
	HD3_TA2	3B	EC	RW	8	00	Transmit Address 2
	HD3_RAW1	3C	F0	RW	8	00	Receive Address Wildcard 1
	HD3_RAW2	3D	F4	RW	8	00	Receive Address Wildcard 2
	HD3_RA1	3E	F8	RW	8	00	Receive Address Register 1
	HD3_RA2	3F	FC	RW	8	00	Receive Address Register 2
•	HD1_SRC	40	100	RW	8	X	Source data: [1:0]: line selection (0.. 2: U-S-A) [4:2]: burst selection (0 … 7) [6:5]: channel selection (0.. 2: B1-B2-D)
•	HD2_SRC	41	104	RW	8	X	Source data: [1:0]: line selection (0.. 2: U-S-A) [4:2]: burst selection (0…7) [6:5]: channel selection (0.. 2: B1-B2-D)
•	HD3_SRC	42	108	RW	8	X	Source data: [1:0]: line selection (0.. 2: U-S-A) [4:2]: burst selection (0…7) [6:5]: channel selection (0.. 2: B1-B2-D)
•	HD_BREV	43	10C	RW	3	0	bit-reverse data bit (lsb fl‡ msb) read / written from / to the HDLC Fifo bit[0]=1 bit-reverse data from / to HDLC 1 bit[1]=1 bit-reverse data from / to HDLC 2 bit[2]=1 bit-reverse data from / to HDLC 3 bit[4]=1 bit reversed per block of 2 for D channel transmission through HDLC1 in transparent mode bit[5]=1 bit reversed per block of 2 for D channel transmission through HDLC2 in transparent mode bit[6]=1 bit reversed per block of 2 for D channel transmission through HDLC3 in transparent mode

	Name	Addr.		Access	Width(bits)	Reset(hex)	Function
		word	byte				
•	HD1_TX	44	110	R	8	X	HDLC packet ready to be transmitted in the current frame
•	HD2_TX	45	114	R	8	X	HDLC packet ready to be transmitted in the current frame
•	HD3_TX	46	118	R	8	X	HDLC packet ready to be transmitted in the current frame
UART							
	UART_RBR	50(*)	140	R	8	00	Receiver Buffer Register
	UART_THR	50(*)	140	W	8	00	Transmitter Holding Register
	UART_IER	51(*)	144	RW	8	00	Interrupt Enable Register
	UART_IIR	52	148	R	8	00	Interrupt Ident Register
	UART_FCR	52	148	W	8	00	Fifo Control Register
	UART_LCR	53	14C	RW	8	00	Line Control Register (with UART_LCR[7]=DLAB bit)
	UART_MCR	54	150	RW	8	00	Modem Control Register
	UART_LSR	55	154	R	8	60	Line Status Register
	UART_MSR	56	158	RW	8	00	Modem Status Register
	UART_SCR	57	15C	RW	8	00	Scratch Register
	UART_DLL	50(*)	140	RW	8	00	Divisor Latch Register (LSB)
	UART_DLM	51(*)	144	RW	8	00	Divisor Latch Register (MSB) (*) Register addresses 50 & 51 corresponds to - DLL & DLM ONLY if UART_LCR[7]=DLAB=1 - RBR/THR & IER if UART_LCR[7]=DLAB=0
GCI router							
•	Ui_B1_CPU	60	180	RW	8	X	CPU value for the B1 upstream Ui channel
•	Ui_B2_CPU	61	184	RW	8	X	CPU value for the B2 upstream Ui channel
•	Ui_M_CPU	62	188	RW	8	X	CPU value for the M upstream Ui channel
•	Ui_E_CPU	63	18C	RW	8	X	CPU value for the C/I byte upstream Ui channel bit[7:6] = D bits bit[5:2] = CI bits bit[1:0] = AE bits
•	Si_B1_CPU	64	190	RW	8	X	CPU value for the B1 downstream Si channel
•	Si_B2_CPU	65	194	RW	8	X	CPU value for the B2 downstream Si channel
•	Si_M_CPU	66	198	RW	8	X	CPU value for the M downstream Si channel
•	Si_E_CPU	67	19C	RW	8	X	CPU value for the C/I byte downstream Si channel bit[7:6] = D bits bit[5:2] = CI bits bit[1:0] = AE bits
•	Ai_B1_CPU	68	1A0	RW	8	X	CPU value for the B1 downstream Ai channel
•	Ai_B2_CPU	69	1A4	RW	8	X	CPU value for the B2 downstream Ai channel
•	Ai_M_CPU	6A	1A8	RW	8	X	CPU value for the M downstream Ai channel
•	Ai_E_CPU	6B	1AC	RW	8	X	CPU value for the C/I byte downstream Ai channel bit[7:6] = D bits bit[5:2] = CI bits bit[1:0] = AE bits
•	Ui_B1_RX	6C	1B0	R	8	X	B1 channel read from the Ui downstream GCI frame
•	Ui_B2_RX	6D	1B4	R	8	X	B2 channel read from the Ui downstream GCI frame
•	Ui_M_RX	6E	1B8	R	8	X	M channel read from the Ui downstream GCI frame
•	Ui_E_RX	6F	1BC	R	8	X	[D,CI,AE] channel read from the Ui down GCI frame
•	Si_B1_RX	70	1C0	R	8	X	B1 channel read from the Si upstream GCI frame

	Name	Addr.		Access	Width(bits)	Reset(hex)	Function
		word	byte				
•	Si_B2_RX	71	1C4	R	8	X	B2 channel read from the Si upstream GCI frame
•	Si_M_RX	72	1C8	R	8	X	M channel read from the Si upstream GCI frame
•	Si_E_RX	73	1CC	R	8	X	[D,CI,AE] channel read from the S up GCI frame
•	Ai_B1_RX	74	1D0	R	8	X	B1 channel read from the Ai upstream GCI frame
•	Ai_B2_RX	75	1D4	R	8	X	B2 channel read from the Ai upstream GCI frame
•	Ai_M_RX	76	1D8	R	8	X	M channel read from the Ai upstream GCI frame
•	Ai_E_RX	77	1DC	R	8	X	[D,CI,AE] channel read from the Ai upstream GCI frame
•	GCI_CPU_RX	78	1E0	W	8	0	Specify the burst targeted by the ARM CPU registers access Rx registers access bit[1:0] = target line U-S-A bit[4:2] = CPU target burst bit[7:5] = Rx target burst
•	GCI_ITU	79	1E4	RW	8	00	bit[i]: CI activity detected on U, burst I Read bit[I]=1: activity detection; Write bit[I]=1: interrupt cleared
•	GCI_ITS	7A	1E8	RW	8	00	bit[i]: CI activity detected on S, burst i
•	GCI_ITA	7B	1EC	RW	8	00	bit[i]: CI activity detected on A, burst i
•	GCI_MSKU	7C	1F0	RW	8	FF	bit[i]=1: mask CI activity detected on U, burst i
•	GCI_MSKS	7D	1F4	RW	8	FF	bit[i]=1: mask CI activity detected on S, burst i
•	GCI_MSKA	7E	1F8	RW	8	FF	bit[i]=1: mask CI activity detected on A, burst i
•	GCI_SRC_BUR	7F	1FC	RW	3	0	Target burst selection for reading _SRC registers
•	U_B1_SRC	80	200	RW	8	X	Source control register for target U_B1 upstream: [7:5] source burst [4:3] source line [2:0] target burst
•	U_B2_SRC	81	204	RW	8	X	Source control register for target U_B2 upstream
•	U_D_SRC	82	208	RW	8	X	Source control register for target U_D upstream
•	U_CI_SRC	83	20C	RW	8	X	Source control register for target U_CI upstream
•	U_M_SRC	84	210	RW	8	X	Source control register for target U_M upstream
•	U_AE_SRC	85	214	RW	8	X	Source control register for target U_AE upstream
•	S_B1_SRC	86	218	RW	8	X	Source control register for target S_B1 downstream
•	S_B2_SRC	87	21C	RW	8	X	Source control register for target S_B2 downstream
•	S_D_SRC	88	220	RW	8	X	Source control register for target S_D downstream
•	S_CI_SRC	89	224	RW	8	X	Source control register for target S_CI downstream
•	S_M_SRC	8A	228	RW	8	X	Source control register for target S_M downstream
•	S_AE_SRC	8B	22C	RW	8	X	Source control register for target S_AE downstream
•	A_B1_SRC	8C	230	RW	8	X	Source control register for target A_B1 downstream
•	A_B2_SRC	8D	234	RW	8	X	Source control register for target A_B2 downstream
•	A_D_SRC	8E	238	RW	8	X	Source control register for target A_D downstream
•	A_CI_SRC	8F	23C	RW	8	X	Source control register for target A_CI downstream
•	A_M_SRC	B0	2C0	RW	8	X	Source control register for target A_M downstream
•	A_AE_SRC	B1	2C4	RW	8	X	Source control register for target A_AE downstream
•	U_B1_SWP	B2	2C8	RW	8	X	bit[i] = swap / no swap for B1 channel on U burst i
•	U_B2_SWP	B3	2CC	RW	8	X	bit[i] = swap / no swap for B2 channel on U burst I
•	S_B1_SWP	B4	2D0	RW	8	X	bit[i] = swap / no swap for B1 channel on S burst I

	Name	Addr.		Access	Width(bits)	Reset(hex)	Function
		word	byte				
•	S_B2_SWP	B5	2D4	RW	8	X	bit[i] = swap / no swap for B2 channel on S burst I
•	A_B1_SWP	B6	2D8	RW	8	X	bit[i] = swap / no swap for B1 channel on A burst I
•	A_B2_SWP	B7	2DC	RW	8	X	bit[i] = swap / no swap for B2 channel on A burst i
Clock Generation							
•	CLK_GCI	90	240	RW	8	00	bit[5:0]: GCI DCL selection per interface. U,S,A: bit[1:0] = for the U GCI router bit[3:2] = for the S GCI router bit[5:4] = for the A GCI router 0 = non-active (default at reset) 1 = master : dclMstr / fscMstr 2 = 512k: dcl512 / fsc512 3 = 4096k: dcl4096 / fsc4096 bit[7:6] : GCI Master FSC selection (fscMstr) 0 = Xtal (default at reset) 1 = U 2 = S 3 = A
•	CLK_1	91	244	RW	8	00	Parameters for CKOUT: bits [7.. 4]: Disabled if equal to "1001" bits [3.. 0]: Freq. Division par. CLK_1 for CKOUT
•	CLK_2	92	248	RW	8	00	Parameters for ARM Clock: bits [7.. 4]: Disabled if equal to "1001" bits [3.. 0]: Freq. Division par. CLK_2 for ARMclock
•	CLK_3	93	24C	RW	8	00	bit [0] : DPLL status (read only) 0 = internal gci clocks not synchronized with the extern reference; dpll is tracking 1 = internal gci clocks synchronized bit [1] : UART clock disable bit [3:2]: APB clock division factor 0 = 15.36 Mhz / 4 (max.speed) 1 = 15.36 Mhz / 8 2 = 15.36 Mhz / 16 3 = 15.36 Mhz / 32
Interrupt							
•	IRQ1_ST	94	250	R	8	00	Interrupt status after mask
•	IRQ1_STR	95	254	R	8	00	Interrupt status without mask
•	IRQ1_EN	96	258	R	8	00	Interrupt mask to be used
•	IRQ1_ENSET	97	25C	W	8	00	Set maskbit control input register
•	IRQ1_ENCLR	98	260	W	8	00	Clear maskbit control input register
•	IRQ1_ACK	99	264	W	8	00	Clear interrupt status register
•	IRQ2_ST	9A	268	R	8	00	Interrupt status after mask
•	IRQ2_STR	9B	26C	R	8	00	Interrupt status without mask
•	IRQ2_EN	9C	270	R	8	00	Interrupt mask to be used
•	IRQ2_ENSET	9D	274	W	8	00	Set maskbit control input register
•	IRQ2_ENCLR	9E	278	W	8	00	Clear maskbit control input register
•	IRQ2_ACK	9F	27C	W	8	00	Clear interrupt status register

Name	Addr. word	Addr. byte		Access	Width(bits)	Reset(hex)	Function
Parallel IO							
• PAB_OUT	A0	280	RW	8	00		bit[7..4]: PA output data register bit[3:0]: PB output data register
• PAB_IN	A1	284	R	8	00		bit[7..4]: PA input data register bit[3:0]: PB input data register
• PAB_DIR	A2	288	RW	8	00		bit[7..4]: PA direction control register (0=input) bit[3:0]: PB direction control register (0=input)
• PCD_IN	A4	290	R	8	00		bit[7..4]: PC input data register bit[3:0]: PD input data register
• PCD_DIR	A5	294	RW	8	00		bit[7..4]: PC direction control register (0=input) bit[3:0]: PD direction control register (0=input)
• PEF_OUT	A6	298	RW	8	00		bit[7..4]: PE output data register bit[3:0]: PF output data register
• PEF_IN	A7	29C	R	8	00		bit[7..4]: PE input data register bit[3:0]: PF input data register
• PEF_DIR	A8	2A0	RW	8	00		bit[7..4]: PE direction control register (0=input) bit[3:0]: PF direction control register (0=input)
GCI Router ext'							
• see before	B0-BF						see GCI Router pages
Timers and Watch-dog							
• TI_1L	C0	900	RW	8	FF		Timer 1 lsb R: counter value W: initial value
• TI_1M	C1	304	RW	8	FF		Timer 1 msb R: counter value W: initial value
• TI_2L	C2	308	RW	8	FF		Timer 2 lsb R: counter value W: initial value
• TI_2M	C3	30C	RW	8	FF		Timer 2 msb R: counter value W: initial value

• TI_CMD	C4	310	RW	8	00		Timer command register:
							Timer 1
							bit[0]: count (=1:counting; =0:not counting)
							bit[1]: init (set to 1 to re-initialise; self-clearing bit)
							bit[2]: fast (=1: fast clock = Xtal / 4;
							=0: slow clock = Xtal / 16)
							bit[3]: capEn (=1:capture enabled; =0:disabled)
							(cleared by HW when captured)
							Timer 2:
							bit[4]: count (=1:counting; =0:not counting)
							bit[5]: init (set to 1 to re-initialise; self-clearing bit)
							bit[6]: fast (=1: fast clock = Xtal / 4;
							=0: slow clock = Xtal / 16)
							bit[7]: / (not used)

•	TI_C1L	C5	314	R	8	FF	Timer 1 capture register : lsb
•	TI_C1M	C6	318	R	8	FF	Timer 1 capture register : msb
•	TI_CSEL	C7	31C	RW	2	0	capture source selection register : 00 : CI change (default) 01 : external interrupt IT 10 : PB input port change 11 : PA input port change

	Name	Address		Access	Width(bits)	Reset(hex)	Function
		word	byte				
•	WD_SC	C8	320	4 RW R	F		Watch-dog status and control register bit[1..0] : Init_value selection 00 : 0.48 sec 01 : 0.61 sec 10 : 0.89 sec 11 : 1.02 sec (default) bit[3..2] : State of watch-dog FSM (Read only) 00 : init & count 01 : unlock1 10 : unlock2 11 : disable (default)

•	WD_MAGIC	C9	324	RW	8	00	Watch-dog magic word register:
							value written : command issued
							value=DC : Kick command;
							value=A5 : Magic1-word command;
							value=5A : Magic2-word command;
							value=AF : Magic3-word command;

D-channel collision avoidance

The pseudo-C code below is intended to show a user of the MTC-20270 or 20280 families of ISDN controllers ("ITC") how to implement the D-channel collision avoidance protocol.

It is based on the use of the MTC-2027x series of ISDN Integrated NT devices, whose S interface support the use of the DE collision protocol. The same applies to the MTC-20172 'S' interface circuit.

The basic algorithm is:

1. Check that the S bus is not using the D channel by monitoring the BUSY bit in the S interface device.
2. Block the use of the S bus D channel by forcing an active condition on the E bit.
3. Switch the upstream D channel to come from the transmitted data stream of one of the MTC-20280's HDLC controllers (e.g. HDLC1).
4. Send the data packet
5. Wait for the packet to end by monitoring the TIF bit, and then allow time for the flag character to leave
6. Switch the upstream U D channel back to the S bus
7. Unblock the S bus D channel by letting the E bit follow the S bus D bit.

The low-level routines d=read_M(p,a) and write_M(p,a,d) read and write data d to/from GCI port p (p=U,S or A) at register address a, using the M-channel protocol.

The ITC register map is externally declared as a struct pointer, which is initialised to the physical hardware address. A simple interrupt routine driven by the GCI frame clock increments an int-sized counter called framecount.

NOTE: When a constant value needs to be used from a CPU register, the value should be written into the CPU register twice in two consecutive GCI frames. (This is an artifact of the architecture with two parallel RAM spaces – the constant value must be written into both RAM spaces. Failure to do so results in the output value alternating between the LAST output value and the new ('constant') value).

```
/* D-channel conflict handling in C */

extern struct *ITC;           /* Defines ITC registers */
extern int framecount;               /* counter incremented each GCI frame */

send_D-chan(char *p)
{
int fc;
        while((read_M(S,0) & BUSY))   {;}
                                      /* Wait until BUSY bit in INT S is clear */
        write_M(S,2,0x30);
                /* Set DE bus active (DE pin is bonded LOW on MTC-2027x ISDN NT chips) */
        ITC->U_DX_SRC = 3;        /* Set Upstream U D-channel to HDLC1 Tx */

        /* SEND DATA PACKET TO HDLC1 HERE */

        while((ITC->HD1-SSTAT & TIF)) {;}
                                             /* Wait until TIF bit goes inactive */
        fc=framecount;                       /* Present framecounter value*/
        while ((framecount-fc) < 4) {;}
                        /* Let 4 complete GCI frames pass to allow flag to leave */
        ITC->U_DX_SRC = 2;       /* Connect upstream U D channel to S */
        write_M(S,2,0x20);
                        /* Disable DE bus to allow S bus D channel to operate */}
```

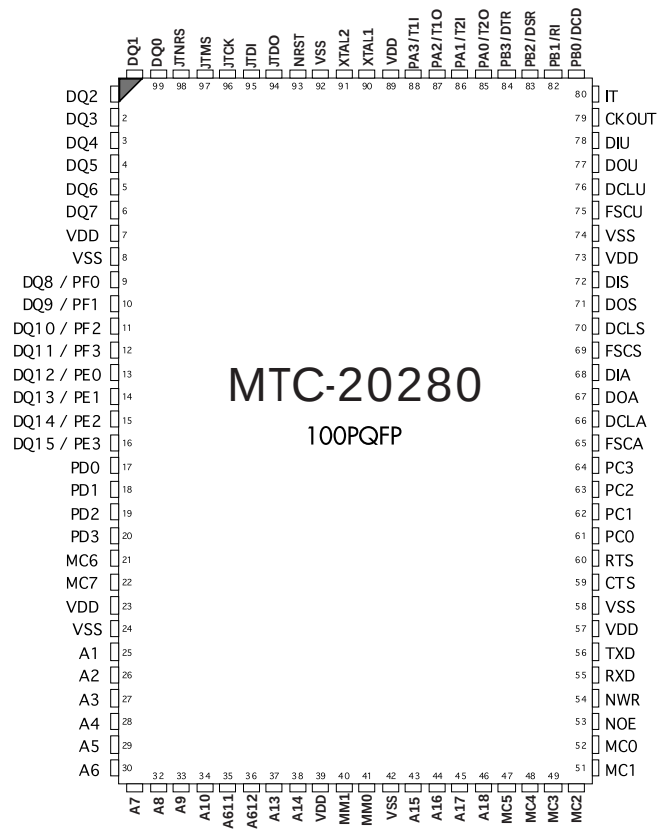


Figure 22 : Device Pinout

Package and Pinout

The device is mounted in a 100-pin PQFP package. The following figures show the pin-out names and package orientations in top view.

Pin description and assignment

The next table lists the device pins and their. The type of pin is encoded with a letter code, such as “DId” for a digital input with internal pull down.

The first letter differentiates between:

- D: Digital
- A: Analog
- P : Power

The second letter differentiates between:

- I : Input
- O: Output
- B : Bidirectional

The next letter differentiates between:

- d : pin with internal pull-down
- u : pin with internal pull-up

The next letter indicates whether the pin is:

- s : pin with Schmitt trigger input

The next letter indicates whether the pin is:

- 5: 5 Volt tolerant input/output

The next letter indicates whether the pin is:

- z : pin with tri-state output

If pins are unused in the application, the tables show whether they must be connected fixed to power (VDD) or ground (GND), or must remain unconnected (DNC).

Important notes on 5 V tolerant pins

Note 1:

The exact meaning is that these cells tolerate 5V INPUT signals, but they do not generate 5V OUTPUT signals. The I/O cell can be either :

tri-stateable (cells of type “*5z”): this is required for driving bus structures in open drain configuration (e.g. GCI data out). In all cases it requires an external pull-up resistor to either 3.3V or 5V depending on which high output level is requested by the application system

non-tristateable (cells of type “*5”): when no bus must be driven and fast transitions are needed (e.g. GCI clocks and UART outputs). No external pull-up is required, but then a 3.3V high output level will be driven

Note 2:

When using a 5V interface, the system should be designed such that no 5V inputs are applied when the 3.3V power supply is not present, as this limits the lifetime of the device.

Pin description table

Pin Identification			
Nr.	Name	Type	
78	DIU	DIs5	
77	DOU	DO5z	
76	DCLU	DBs5	
75	FSCU	DBs5	
72	DIS	DIs5	
71	DOS	DO5z	
70	DCLS	DBs5	TTL
69	FSCS	DBs5	6mA
68	DIA	DIs5	
67	DOA	DO5z	
66	DCLA	DBs5	
65	FSCA	DBs5	
99	DQ0	DBs	Cmos
100	DQ1	DBs	4mA
1	DQ2	DBs	
2	DQ3	DBs	
3	DQ4	DBs	
4	DQ5	DBs	
5	DQ6	DBs	
6	DQ7	DBs	
9	DQ8	DBs	
10	DQ9	DBs	
11	DQ10	DBs	
12	DQ11	DBs	
13	DQ12	DBs	
14	DQ13	DBs	
15	DQ14	DBs	
16	DQ15	DBs	
22	MC7	DOz	
25	A1	DO	Cmos
26	A2	DO	4mA
27	A3	DO	
28	A4	DO	
29	A5	DO	
30	A6	DO	
31	A7	DO	
32	A8	DO	
33	A9	DO	
34	A10	DO	
35	A11	DO	
36	A12	DO	
37	A13	DO	
38	A14	DO	
43	A15	DO	
44	A16	DO	
45	A17	DO	
46	A18	DO	
52	MC0	DO	Cmos
51	MC1	DO	4mA
50	MC2	DO	
49	MC3	DO	
48	MC4	DO	
47	MC5	DO	
21	MC6	DO	
54	NWR	DO	
53	NOE	DO	
40	MM1	DI	Cmos
41	MM0	DI	4mA
55	RXD	DIs5	TTL
56	TXD	DO5	6mA
59	CTS	DIs5	
60	RTS	DO5	
98	NTRST	DId	Cmos
97	TMS	DIu	4mA
96	TCK	DIu	
95	TDI	DIu	
94	TDO	DO	
90	XTAL1	DIs	Cmos
91	XTAL2	DIs	4mA
79	CKOUT	DO	
93	NRST	DB5s	TTL
80	IT	DI5s	6mA
85	PA0	DBs5	TTL
86	PA1	DBs5	6mA
87	PA2	DBs5	
88	PA3	DBs5	
81	PB0	DBs5	TTL
82	PB1	DBs5	6mA
83	PB2	DBs5	
84	PB3	DBs5	
61	PC0	DBs5	TTL
62	PC1	DBs5	6mA
63	PC2	DBs5	
64	PC3	DBs5	
17	PD0	DBs5	TTL
18	PD1	DBs5	6mA
19	PD2	DBs5	
20	PD3	DBs5	
7	VDD	PI	
23	VDD	PI	
39	VDD	PI	
57	VDD	PI	
73	VDD	PI	
89	VDD	PI	
8	VSS	PI	
24	VSS	PI	
42	VSS	PI	
58	VSS	PI	
74	VSS	PI	
92	VSS	PI	

Pin function in normal operating mode

The following table gives a summary of all MTC-20280 pins and their functions in normal (i.e. non-test) mode. Test modes are used by Alcatel for production testing only.

Note that the allocation of GCI ports to U,S, or A (*) is for convenience only: all three ports are functionally identical and can be interchanged.

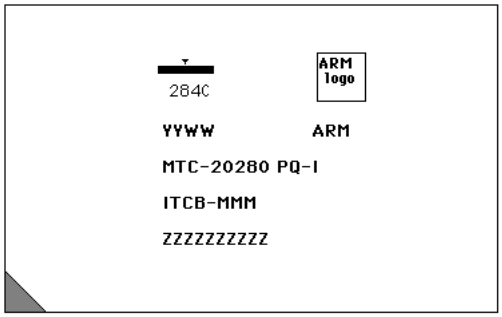
Note also that depending on the configuration in which the MTC-20280 is used:

- the MSB byte of the data bus can be re-used as parallel IO; this is the case when the Word Access Mode is disabled.
- the parallel I/O port PB[3..0] can be used to access input/output control signals of the UART module for implementing a modem communication protocol; this happens as soon as the Pb2Uart mode is selected.
- the parallel I/O port PA[3..0] can be used to access one input and/or one output control signal for each Timer module. The input allows counting based on externally applied rising-edge events; the output shows a toggling signal, based on the timer interrupt events, which can be used as a clock. The connection to either the general purpose Parallel IO function or a Timer function, is programmed on a bit-basis by setting the corresponding CHIP_CFG[i] bit .

Pin Nr.	Pin name		Description
76	DCLU	GCI-U* INT.	GCI Data clock associated with the U-GCI interface
75	FSCU		GCI Frame clock associated with the U-GCI interface
78	DIU		GCI Data from the U-GCI interface
77	DOU		GCI Data to the U-GCI interface
70	DCLS	GCI-S* INT.	GCI Data clock associated with the S-GCI interface
69	FSCS		GCI Frame clock associated with the S-GCI interface
72	DIS		GCI Data from the S-GCI interface
71	DOS		GCI Data to the S-GCI interface
66	DCLA	GCI-A* INT.	GCI Data clock associated with the A-GCI interface
65	FSCA		GCI Frame clock associated with the A-GCI interface
68	DIA		GCI Data from the A-GCI interface
67	DOA		GCI Data to the A-GCI interface

Pin Nr.	Pin name		Description
99,100, 1...16	DQ15..0	RAM	Bidirectional data bus; DQ0:lsb, DQ15:msb ;Msb byte used as parallel ports PE,PF if singleByte access mode is chosen
25...46	A18..1	INT.	Address bus: A18:msb, A1:lsb (note: lsb=MC7 depending on interface mode)
21,22, 47...52	MC7..0		Memory control outputs (meaning dependent on interface mode): e.g., chip select signals (active low), lsb / msb byte selection, and address lsb, depending on MM1..0
40,41	MM1..0		MemoryAccess mode selection
54	NWR		Write enable signal, active low
53	NOE		Output enable signal, active low
55	RXD	UART	Serial data input
56	TXD		Serial data output
59	CTS		Modem control signal : Clear to Send input
60	RTS		Modem control signal : Request to Send output
98	JTNRS	JTAG	JTAG reset signal, active low
97	JTMS		JTAG mode select signal
96	JTCK		JTAG clock (max 10 MHz)
95	JTDI		JTAG data input
94	JTDO		JTAG data output
90	XTAL1	Clocks	Pins to connect an external parallel mode Xtal for the on-chip oscillator. Nominal frequency : 15.36 MHz ±50ppm Remark : XTAL1 may be used as an input for an external master clock source; in that case XTAL2 must be connected to GND
91	XTAL2		
79	CKOUT		Programmable output clock (typically 15.36 MHz for U-interface)
93	NRST		Hardware reset, active low. Schmitt-trigger input with threshold at 1.65 V (CMOS LEVEL). Connect an external (RC) circuit with timing > 1 ms
80	IT		External interrupt source (rising / falling edge or level triggered)
85...88	PA3..0	Parallel IO	Bitwise controlled Input or Output, or potential connection to Timer modules T1 or T2 (PA3=T1I, PA2=T1O, PA1=T2I, PA0=T2O) depending on CHIP_CFG[6..3]
61...64	PC3..0 PD3..0		Bitwise controlled Input or Output
81...84	PB3..0		Bitwise controlled Input or Output, or potential extra connection to the UART for modem communication with PB3 == DTR output, PB2..0 == DSR / RI / DCD inputs depending on CHIP_CFG[0]
7,23,39,57, 73,89	VDD	Power	3.3 V power supply
8,24,42,58 74,92	VSS	Ground	0 V ground

Device branding



Alcatel Microelectronics
info@mie.alcatel.be
http://www.alcatel.com/telecom/micro

Sales Offices

Central Europe
Arabellastraße 4
81925 München
Germany

Tel. +49 89 920 07 70
Fax +49 89 910 15 59

Stuttgart Office
SchwieberdingerStraße 9
70435 Stuttgart
Germany

Tel. +49 711 82145304
Fax +49 711 82144619

Northern Europe
Excelsiorlaan 44-46
1930 Zaventem
Belgium

Tel. +32 2 718 18 11
Fax +32 2 725 37 49

Sales & Design Centres

Italy
Via Trento 30
20059 Vimercate MI
Italy

Tel. +39 0 39 686 4520
Fax +39 0 39 686 6899

Southern Europe
10, rue Latécoère, B.P.57
78140 Vélizy Cedex
France

Tel. +33 1 46 32 53 86
Fax +33 1 46 32 55 68

USA
M/S 412-115
1225 N. Alma Road
Richardson
TX 75081-2206

Tel. +1 972 996 2489
Fax +1 972 996 2503

Representatives

Northern Germany
TRIAS
Moerser Landstraße 408
47802 Krefeld
Germany

Tel. +49 2151 95 30 10
Fax +49 2151 95 30 105

Taiwan ROC
Alcatel ITS Ltd
Suite D, 8th floor 133
Section 3, Min Seng E. Road
Taipei 015, Taiwan

Tel. +886 2 717 1255
Fax +886 2 717 1250

Japan
Alcatel ITS Japan Ltd
Yubisu Garden Plae Tower 24
20-3 Ebisu 4-chrome, Shibuyaku
Tokyo, 150
Japan

Tel. +81 3 5424 8561
Fax +81 3 5424 8581

Marketing & Design Centre

Excelsiorlaan 44-46
1930 Zaventem
Belgium

Tel. +32 2 718 18 11
Fax +32 2 725 37 49

Headquarters Manufacturing & Customer Service

Westerring 15
9700 Oudenaarde
Belgium

Tel. +32 55 33 24 70
Fax +32 55 33 27 68

USA

Premier Technical Sales
SAN DIEGO, CA, 2011 Via Tiempo, Cardiff, CA 92007, Tel. (760) 943 6222, Fax (760) 943 0425
SANTA CLARA, CA, 3235 Kifer Road, Suite 110, Santa Clara, CA 95051, Tel. (408) 736 2260, Fax (408) 736 2826
TUSTIN, CA, 2660 Walnut Ave, Unit H, Tustin, CA 92680, Tel. (714) 573 8242, Fax (714) 573 4942
ATLANTA, GA, 135 Arden Way, Peachtree City, GA 30269, Tel. (770) 632 9648, Fax (770) 486 4098
BOSTON, MA, 33 Boston Post Road West, Suite 270, Marlboro, MA 01752, Tel. (508) 460 1730, Fax (508) 460 1731
PORTLAND, OR, 5319 S.W. Westgate Dr. Suite 136, Portland, OR 97221, Tel. (503) 297 3956, Fax (503) 297 4956
AUSTIN, TX, 12148 Jollyville Road, Unit 422, Austin, TX 78759, Tel. (512) 257 8218, Fax (512) 257 1714
DALLAS, TX, 800 East Campbell Road, Suite 199, Richardson, TX 75081, Tel. (972) 680 5233, Fax (972) 680 5234
OTTAWA, ONTARIO, CANADA, 43 Pretty Street, Stittsville, Ontario, K2S1A4 Canada, Tel. (613) 836 1779, Fax (613) 836 4459

12/98· DS0269b