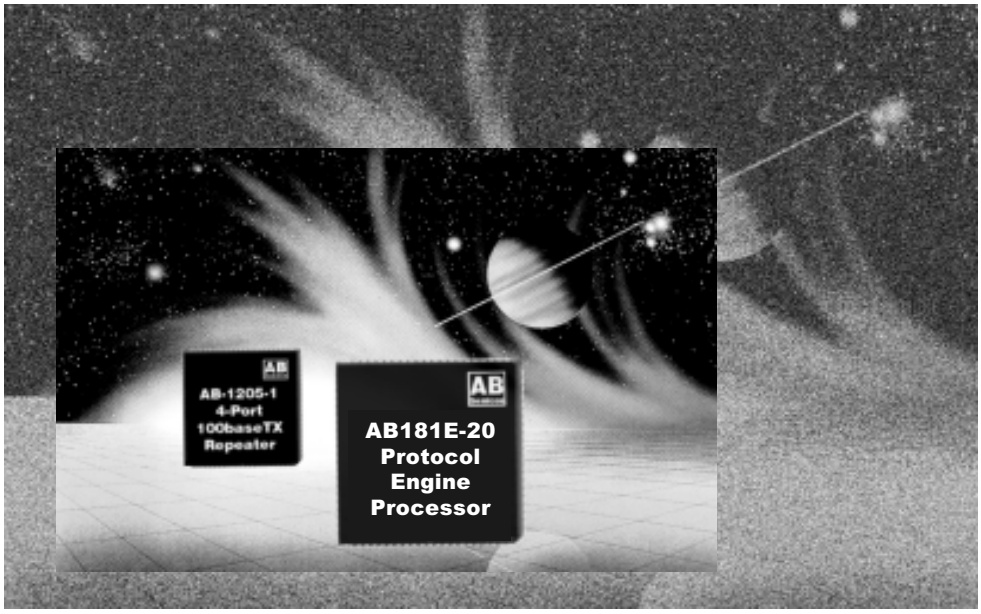
AB181E-20™

8-bit Enhanced OCA Processor

for General Purpose, Protocol Engines and Robotics Applications

Combines High Performance and Low-Cost



Product Specification

AB Semicon AB181E-20™

General Purpose and Protocol Engine Processor
Product Specification

For the latest information on the AB181E-20, check the product specification on the AB Semicon web-site at:

http://www.ab-semicon.com

Disclaimer

AB Semicon Limited reserves the right to revise this publication and to make changes from time to time to the contents hereof without obligation to notify any person or organization of such revision or changes. AB Semicon Limited has endeavoured to ensure that the information in this publication is correct, but will not accept liability for any error or omission.

AB181E-20™

General Purpose and Protocol Engine One Cycle Architecture Processor

The answer to every Z80, Z180, HD64180 user in the world - yes it has the horsepower you were looking for, no you do not have to re-write your code, you can use your existing Z80 Assemblers, Linkers and C-compilers. Some minor differences do exist.

Features:

* 40 MHz frequency synthesized 8 bit processor
* Memory to Memory Block Transfer at 10 Mbytes/sec
* Memory to I/O and I/O to Memory Block Transfer at 10 Mbytes/sec
* 8 bit Data bus
* 20 bit Address bus
* Synchronous serial I/O suitable for Apple Local Talk up to 1Mbaud
* Asynchronous Serial I/O up to 2Mbaud suitable for IrDA up to 2Mbit/s
* Fixed point 32 bit arithmetic unit
* 100pin Quad Flat Pack packaging
* Two 16bit Timers
* Dual 3.3/5V operation or single 3.3V
* Each Clock cycle (at 20 MHz) one instruction (for single byte instructions) or the instruction is carried out at the end of the last byte fetch of a multi-byte instruction

Contents

Appendices

Introduction

This guide provides product information and specifications for the AB181E-20 General Purpose and Protocol Engine Processor, detailing the performance capability of this fast, state-of-the-art microprocessor using AB Semicon's unique OCA (One Cycle Architecture) technology.

Applications include:

Network Connected Devices
Digital Cameras
Cell Phones
Automotive
Motor Controllers
Digital Signal Processing (up to 500 kHz)
Robotics/Motion Control

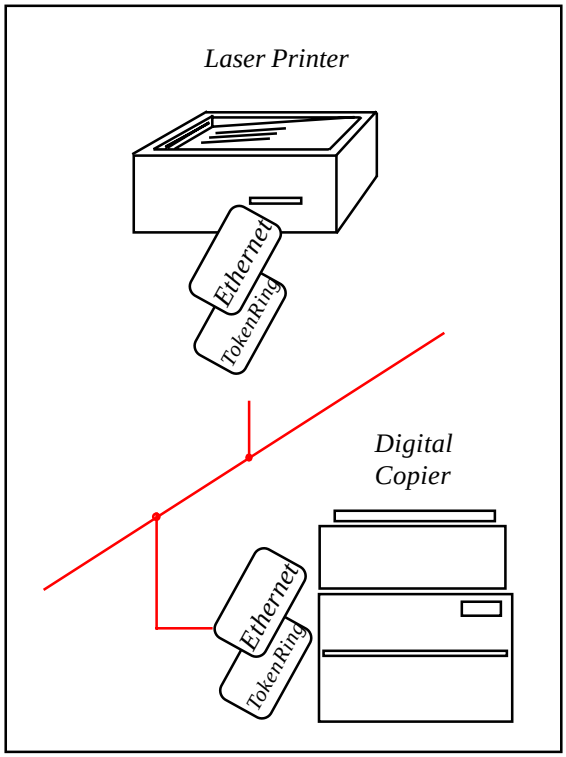
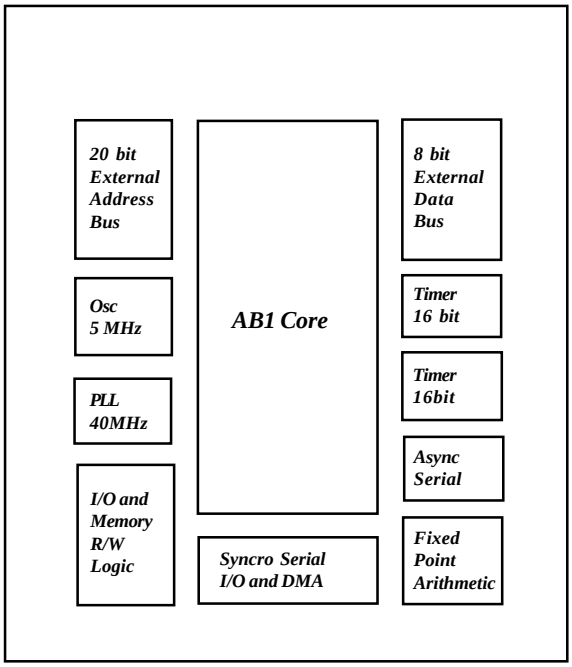


Figure 1

Chip Structure

8 bit processor core

The processor is driven by a 5MHz crystal oscillator and phase locked loop (PLL) which generates the on chip 40MHz clock. The address bus is connected to the outside world giving a 20bit externally addressable RAM or ROM space. Two DMA channels Memory to Memory and Memory to I/O and I/O to Memory are available. The processor wakes up in 4 Wait State Mode allowing lower speed external ROM to be used.



Functional Blocks *Figure 2*

Applications

The chip can be used in many different applications but is most powerful for protocol engine applications such as Network Controller or Fax modem controller for multi function Copiers, Laser Printers, BubbleJet Printers and Fax Machines. The fixed point arithmetic unit makes the AB181 ideal for use in robotics and other closed-loop servo systems.

In an Application where the existing Processor is already fully utilised this low cost Protocol Engine Processor can take over all the work of the handling of Network Protocol Stacks and provide the raw data to the printer controller. It can also handle the SNMP data to and from the Printer Controller or NPMP™ management information without taking up any of the Main Printer Controllers time.

Digital Copier Application / Multi Function Printer

The AB181E-20 is a very good match for Digital Copier Designs or Multi Function Printer/Scanner Designs. It allows Network or Modem connectivity to be achieved at extremely low hardware cost and board space. The following diagram shows a typical Digital Copier Design making use of an AB181E-20 processor.

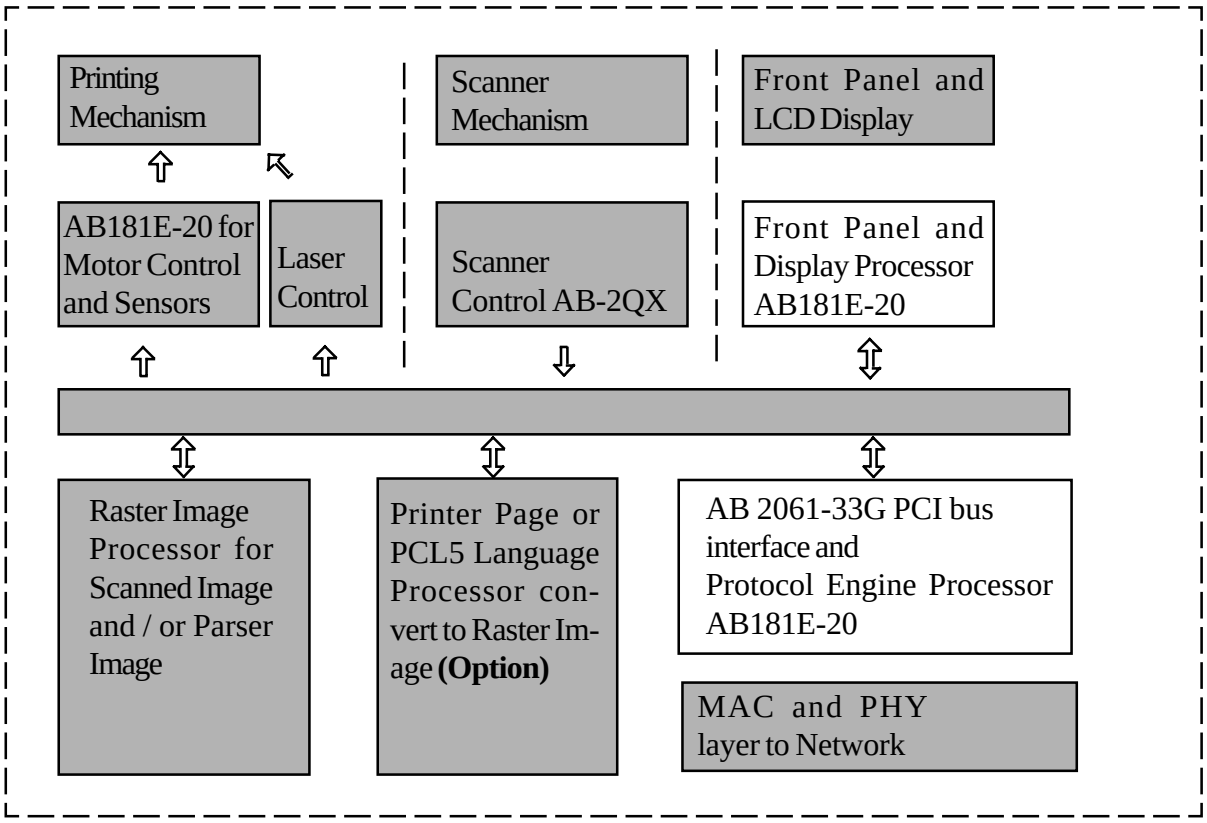


Figure 3

Programmer Guide

Operationally, the AB181E-20 is compatible with the classic Z80 CPU. The programmer has the same register sets, and uses the same op-codes. With a few exceptions (noted later), any program that is designed to run on the Z80 will function on the AB181E-20 with little or no modification. In addition to the Z80 instruction set, the AB181E-20 has some extra instructions designed to increase efficiency in many common applications.

As well as the CPU core, the AB181E-20 also has some on-chip peripherals that are easily accessed by the programmer, and makes the device very powerful in most applications without the need for the plethora of support logic usually required by the Z80. These peripherals are code and functionally compatible with the "180" processors manufactured by Zilog and Hitachi.

The peripherals included on-chip are:

1) A versatile wait-state generator that may be programmed to insert different wait-states in up to 3 different areas of memory, in order to allow different speed memories to be used.

2) A Memory Management Unit (MMU) that allows the total addressed memory range to be up to 1048576 bytes (20 bit addressing).

3) 2 DMA channels, allowing memory-memory and memory-I/O transfers in background.

4) 2 clock-timers, which may be programmed to interrupt the CPU at regular intervals and/or generate a hardware signal.

5) 2 Asynchronous Serial Communications ports that allow 2 RS232 I/O channels, either internally or externally clocked.

6) 1 synchronous serial Communications port, to allow inter-processor communication. Also suitable for downloading many popular FPGA devices.

7) A 4 channel interrupt handler.

8) A DRAM refresh circuit (same as the Z80). This may be disabled for high-speed operation.

9) Afixed-point arithmetic unit optimised for control applications.

The following is a list of exceptions to the Z80 instruction set that may need to be taken into account when porting code:

1) Not all Z80 interrupt modes are supported. The AB181E-20 uses interrupt mode 2 only. The interrupt mode instructions are treated as NOPs. Z80 interrupt acknowledge cycles are not generated.

2) Due to the fast internal state machine, software timing loops will execute much faster than on a Z80. This may need to be taken into account with some programs.

3) The AB181E-20 powers up with maximum wait states, and DRAM refresh cycles enabled. To obtain maximum performance some additional instructions should be inserted during the initialisation phase to set the optimum configuration.

4) The Z80 has some undocumented instructions, such as being able to load the upper or lower 8 bits of the index register (IY and IX) independently. This was used by a handful of programs, especially some "copy protected" code. The AB181E-20 does not support all of the undocumented Z80 instructions.

Hardware Guide

The AB181E-20 interface signals follow the classic Z80 conventions, and interfacing is straightforward using conventional logic, or configurable logic devices. Internally, the CPU state-machine has been greatly enhanced to give very fast code execution times. Most instructions have no internal "T" states, and the AB181E-20 (without wait-states) will execute based on 1 memory access per internal clock cycle. The internal clock runs at 4 times the crystal speed. This internal clock is available on an output pin.

Due to this difference, and the fact that conventional Z80 peripherals cannot be used (they are too slow), the Z80 hardware state signals E, ST, LIR are not provided. This should not present any down-sides, and the AB181E-20 may be used in most applications as a replacement for the 180 processors. Due to the internal wait-state generator, fast memory is not a necessity, although faster execution times are naturally available if wait-states can be avoided. With 1 wait-state, there is at least a three-fold increase in speed over the fastest 180 processor currently available, and a five-fold increase is usual with faster memory and no wait-states. Some applications will see an even more dramatic improvement, because the "block" instructions (e.g. LDIR) are greatly enhanced.

The external -WAIT- input has different timing to that of the Z80 or the 180. Generally this makes interfacing easier, as -WAIT- may be asserted together with -IOE- or -ME-, without needing to gate -RD- or-WR- signals. It is a consideration that may trip the unwary however.

The core logic of the AB181E-20 operates at 3.3V, and so this must be provided. The I/O pins are capable of either 3.3V or true 5V operation, depending upon what voltage is applied to the I/O power supply pins. Thus the AB181E-20 is suitable for use in either 3.3V or 5V circuits. For 5V use, a simple zener-diode voltage dropping circuit is ample to supply the low-current core.

The I/O bus operates at 2 times the core frequency (see timing diagrams) which is 8 times the crystal frequency.

Device Details

Packaging Information

The following diagram shows the packaging for the AB181E-20 Protocol Engine Processor:

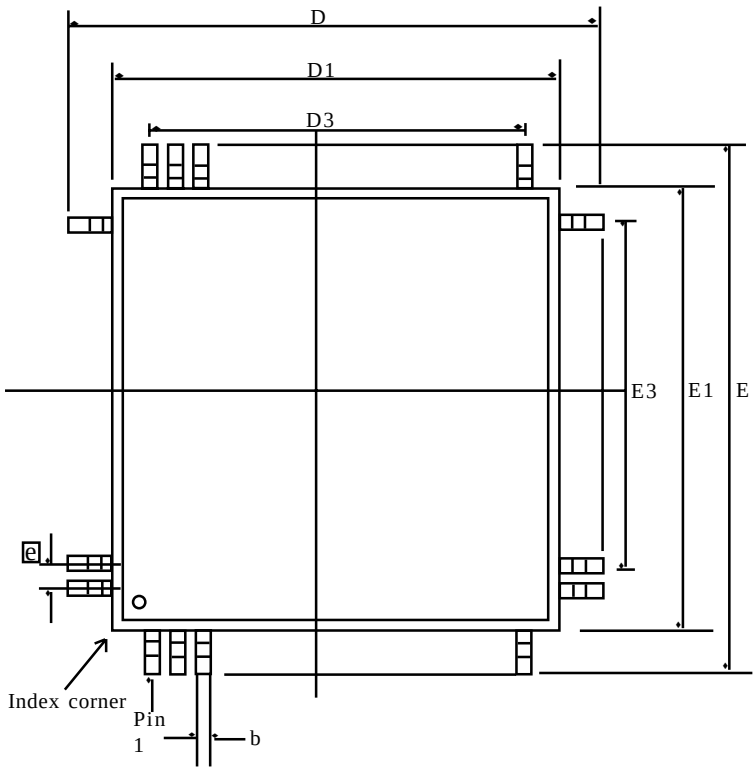


Figure 4

Packaging Information

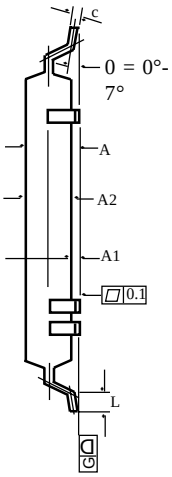


Figure 5

Symbol	Control Dimensions in millimetres			Alternative Dimensions in inches		
	MIN	Nominal	MAX	MIN	Nominal	MAX
A	2.80		3.40	0.110		0.134
A1	0.25		0.85	0.010		0.033
A2	2.55		3.05	0.100		0.120
D	23.65		24.15	0.931		0.951
D1	19.80		20.20	0.780		0.795
D3	18.85 REF.			0.742 REF.		
E	17.65		18.15	0.695		0.715
E1	13.80		14.20	0.543		0.559
E3	12.35 REF.			0.486 REF.		
L	0.73		1.03	0.029		0.041
e	0.65 BSC.			0.026 BSC.		
b	0.22		0.38	0.009		0.015
c	0.11		0.23	0.004		0.009
	Pin features					
N	100					
ND	30					
NE	20					
NOTE	RECTANGULAR					

Conforms to JEDEC MO-112 CC-1 Iss. B.

◆ Note: This package is rectangular

I/O Pin Assignment

Pin ID	*Signal*	*Designator*
1	-	*Vss 0V*
2	-	*N/C*
3	-	*N/C*
4	-	*Vss 0V*
5	-	*Vss 0V*
6	*input*	*NMI*
7	*Output*	*A<0>*
8	*Output*	*A<1>*
9	*Output*	*A<2>*
10	*Output*	*A<3>*
11	*Output*	*A<4>*
12	*Output*	*A<5>*
13	*Output*	*A<6>*
14	*Output*	*A<7>*
15	-	*MixVdd 3.3/5V*
16	*Output*	*A<8>*
17	*Output*	*A<9>*
18	-	*N/C*
19	*Output*	*A<10>*
20	*Output*	*A<11>*
21	*Output*	*A<12>*
22	*Output*	*A<13>*
23	*Output*	*A<14>*
24	*Output*	*A<15>*
25	*Output*	*A<16>*

I/O Pin Assignment

Pin ID	*Signal*	*Designator*
26	-	*MixVdd 3.3/5V*
27	-	*MixVdd 3.3/5V*
28	-	*N/C*
29	-	*N/C*
30	-	*MixVdd 3.3/5V*
31	-	*MixVdd 3.3/5V*
32	*Output*	*A<19> /TOUT*
33	*Input/Output*	*D<0>*
34	*Input/Output*	*D<1>*
35	*Input/Output*	*D<2>*
36	*Input/Output*	*D<3>*
37	*Input/Output*	*D<4>*
38	-	*Vss 0V*
39	*Input/Output*	*D<5>*
40	*Input/Output*	*D<6>*
41	*Input/Output*	*D<7>*
42	-	*Vss 0V*
43	*Output*	*RTSO*
44	*Input*	*CTSO*
45	*Input*	*DCDO*
46	*Output*	*TXAO*
47	*Output*	*A<17>*
48	*Output*	*A<18>*
49	-	*MixVdd 3.3/5V*
50	-	*MixVdd 3.3/5V*

I/O Pin Assignment

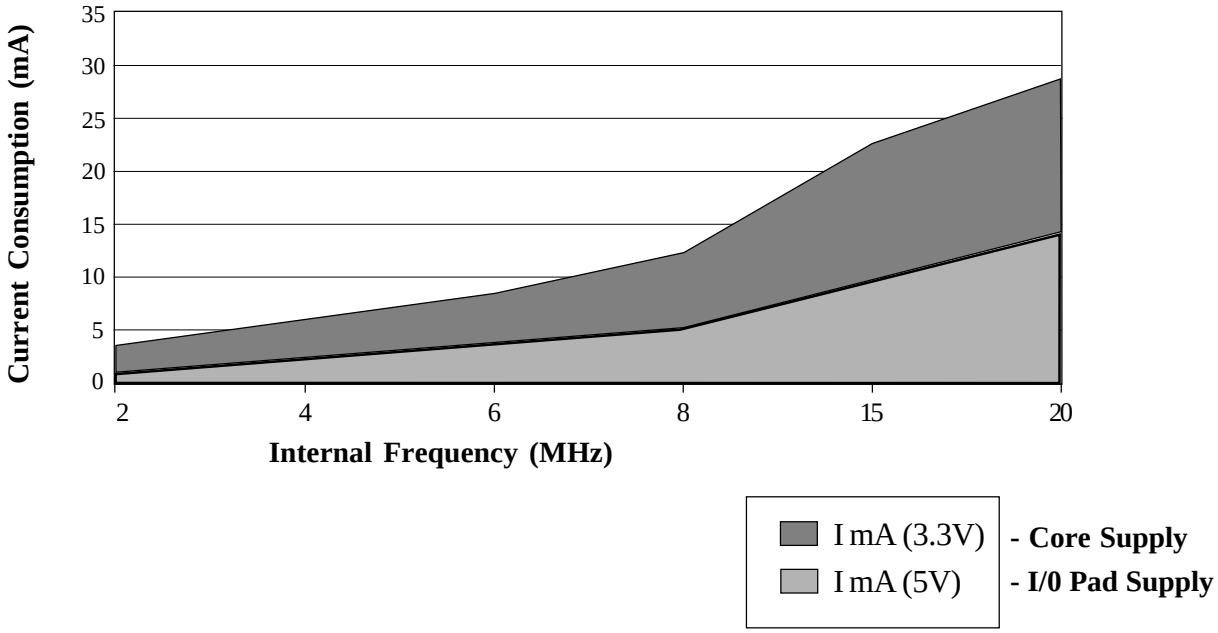
Pin ID	*Signal*	*Designator*
51	-	*N/C*
52	-	*N/C*
53	-	*N/C*
54	-	*MixVdd 3.3/5V*
55	-	*MixVdd 3.3/5V*
56	*Input*	*RXAO*
57	*Input/Output*	*CKAO/DREQO*
58	*Output*	*TXA1*
59	---	*N/C*
60	*Input*	*RXA1*
61	*Input/Output*	*CKA1/TENDO*
62	*Output*	*TXS*
63	*Input*	*RXS/CTS1*
64	-	*Vss 0V*
65	-	*MixVdd 3.3/5V*
66	*Output*	*CKS*
67	*Input*	*DREQ1*
68	*Output*	*TEND1*
69	*Output*	*HALT*
70	*Output*	*REF*
71	*Output*	*IOE*
72	*Output*	*WR*
73	*Output*	*RD*
74	*Output*	*ME*
75	-	*N/C*

I/O Pin Assignment

Pin ID	*Signal*	*Designator*
76	-	*MixVss 0V*
77	-	*MixVss 0V*
78	-	*N/C*
79	-	*N/C*
80	-	*MixVss 0V*
81	-	*MixVss 0V*
82	*Output*	*PHY (ø)*
83	*Input*	*INTO*
84	*Input*	*INT1*
85	*Input*	*INT2*
86	-	*MixVdd 3.3/5V*
87	-	*Vdd3 3.3V*
88	-	*Vdd3 3.3V*
89	-	*N/C*
90	-	*N/C*
91	-	*N/C*
92	*Input*	*LFT (3V sig)*
93	*Input*	*EXTAL (3V sig)*
94	*Output*	*XTAL (3V sig)*
95	-	*Vss 0V*
96	*Input*	*WAIT*
97	*Output*	*BUSACK*
98	*Input*	*RESET*
99	*Input*	*BUSREQ*
100	-	*Vss 0V*

Notes:
1) Connect all Vss and MixVss pins to 0V.
2) Connect all Vdd 3 pins to 3.3V.
3) All MixVdd pins must be connected to the same voltage (either 5V or 3.3V).
4) The voltage on the MixVdd pins determines the operating voltage of all I/O pins.
5) Regardless of the voltage on MixVdd, EXTAL and all test pins are 3.3V inputs only.

Current Consumption v Frequency for the AB181E-20 IC



Frequency	I mA (5V)	I mA (3V3)	V (5V)	V (3V3)
2	1.39	2.08	5	3.28
4	2.6	3.4	5	3.28
6	3.43	4.44	5	3.28
8	5.4	6.65	4.98	3.28
15	9.99	12.35	5.01	3.28
20	13.26	15.72	4.93	3.196

Overview

AB181E-20 Architecture

The AB181E-20 CPU has five functional blocks:

Central Processing Unit: The AB181E-20 uses a superset of the Z80 CPU instruction set, with interface signals that follow Z80 conventions, using conventional logic or configurable logic devices. The CPU has been engineered to give very fast code execution times. In addition to the Z80 instruction set, the AB181E-20 includes extra instructions to increase efficiency in many common applications.

Clock Generator: The clock generator consists of a 5 MHz crystal oscillator and Phased locked loop (PLL) which generates the on-chip 40MHz and 20MHz clocks.

Bus State Controller: This performs all of the bus control and status activity associated with the CPU and on-chip peripherals, including DMA bus exchanges, reset cycles, DRAM refresh, and wait state timing.

Memory Management Unit: The MMU allows the user to increase the available memory of the CPU from 64k (for the Z80) to 1M Byte. This is achieved with a *common / banked area* structure.

Interrupt Controller: In order to provide the correct responses from the CPU, the interrupt controller monitors and prioritizes the various external and internal interrupts and traps.

The AB181E-20 CPU has five on-chip peripherals:

Asynchronous Serial Communications Interface (ASCI - two channels):
The ASCI provides two full-duplex UARTs. Each channel includes a programmable baud rate generator and modem control signals.

DMA Controller: The DMA controller provides high speed transfers from memory to memory, memory to/from I/O, and I/O to I/O. It supports the modes *request* and *cycle steal*. DMA transfers can access the complete 1Mbyte addressing range.

Programmable Reload Timer (PRT - two channels):
The PRT consists of two separate channels containing a 16 bit timer and count reload register. Before reaching the counters, the system clock provides the time base required. Channel 1 has an optional output that enables waveform generation.

Clock Serial I/O (CSIO): The CSIO channel enables synchronous high-speed data communication with other microprocessors, microcomputers and peripherals using a half-duplex serial transmitter and receiver.

Fixed Point Arithmetic Unit: The FPAU is designed to provide functionality that aids the control of servo motors in robotics applications. It also provides a 32-bit multiply-accumulate function.

Internal I/O Registers

The internal I/O registers of the AB180-20 occupy the 64 addresses between 00h and 3Fh on reset. To avoid conflicts with external devices these registers can be relocated within the bottom 256 bytes of the I/O address space.

I/O Control Registers

IOA7	IOA6	Ø	-	-	-	-	-

IOA [7:6] I/O Address Relocation

These bits relocate the internal I/O registers as shown below in Table 1:

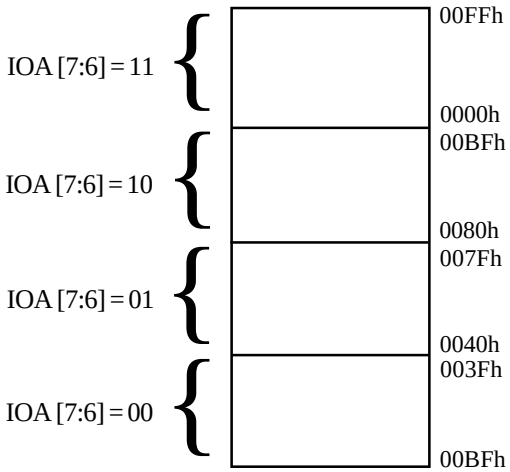


Table 1

Table of Registers

	Register	Mnemonic	Address	
			Binary	Hexadecimal
ASCI	ASCI Control Register A Ch 0	CNTLA0	XX000000	00H
	ASCI Control Register A Ch 1	CNTLA1	XX000001	01H
	ASCI Control Register B Ch 0	CNTLB0	XX000010	02H
	ASCI Control Register B Ch 1	CNTLB1	XX000011	03H
	ASCI Status Register Ch 0	STAT0	XX000100	04H
	ASCI Status Register Ch 1	STAT1	XX000101	05H
	ASCI Transmit Data Register Ch 0	TDR0	XX000110	06H
	ASCI Transmit Data Register Ch 1	TDR1	XX000111	07H
	ASCI Receive Data Register Ch 0	RDR0	XX001000	08H
	ASCI Receive Data Register Ch 1	RDR1	XX001001	09H
CSI/0	CSI/O Control Register	CNTR	XX001010	0AH
	CSI/O Transmit/Receive Data Register	TRDR	XX001011	0BH
Timer	Timer Data Register Ch OL	TMDROL	XX001100	0CH
	Timer Data Register Ch OH	TMDROH	XX001101	0DH
	Reload Register Ch OL	RLDROL	XX001110	0EH
	Reload Register Ch OH	RLDROH	XX001111	0FH
	Timer Control Register	TCR	XX010000	10H
	Reserved		XX010001 ~ XX010011	11H ~ 13H
	Timer Data Register Ch 1L	TMDR1L	XX010100	14H
	Timer Data Register Ch 1H	TMDR1H	XX010101	15H
	Reload Register Ch 1L	RLDR1L	XX010110	16H
	Reload Register Ch 1H	RLDR1H	XX010111	17H
ASCI	Baud Rate Prescaler	PRSCALE	XX011000	18H
Unused	Reserved		XX011001 ~ XX011111	19H ~ 1FH

	Register	Mnemonic	Address	
			Binary	Hexadecimal
DMA	DMA Source Address Register Ch OL	SAROL	XX100000	20H
	DMA Source Address Register Ch OH	SAROH	XX100001	21H
	DMA Source Address Register Ch OB	SAROB	XX100010	22H
	DMA Destination Address Register Ch OL	DAROL	XX100011	23H
	DMA Destination Address Register Ch OH	DAROH	XX100100	24H
	DMA Destination Address Register Ch OH	DAROB	XX100101	25H
	DMA Byte Count Register Ch 0L	BCROL	XX100110	26H
	DMA Byte Register Ch 0H	BCROH	XX100111	27H
	DMA Memory Address Register Ch 1L	MAR1L	XX101000	28H
	DMA Memory Address Register Ch 1H	MAR1H	XX101001	29H
	DMA Memory Address Register Ch 1B	MAR1B	XX101010	2AH
	DMA I/O Address Register Ch 1L	IAR1L	XX101011	2BH
	DMA I/O Address Register Ch 1H	IAR1H	XX101100	2CH
	Internal Wait Control Register	IMWR	XX101101	2DH
	DMA Byte Count Register Ch 1H	BCR1H	XX101111	2FH
	DMA Status Register	DSTAT	XX110000	30H
	DMA Mode Register	DMODE	XX110001	31H
	DMA/ WAIT Control Register	DCNTL	XX110010	32H
INT	IL Register (INT VEctor Register)	IL	XX110011	33H
	INT/ TRAP Control Register	ITC	XX110100	34H
	Reserved		XX110101	35H
Refresh	Refresh Control Register	RCR	XX110110	36H
	Reserved		XX110111	37H
MMU	MMU Common Base Register	CBR	XX111000	38H
	MMU Bank Base Register	BBR	XX111001	39H
	MMU Common/Bank Area Register	CBAR	XX111010	3AH
FPAU	Register Access Port_LOW BYTE	RP_LO	XX111011	3BH
	Register Access Port_HIGH BYTE	RP_HI	XX111100	3CH
	Control/Status Register	AUCNTRL	XX111101	3DH

	Register	Mnemonic	Address	
			Binary	Hexadecimal
	Reserved		XX111110	3EH
I/O	I/O Control Register	ICR	XX111111	3FH

I/O Addressing Notes

The on-chip register addresses are located in the I/O address space from 0000H to 00FFH (16-bit I/O addresses). In order to access the on-chip I/O registers (using I/O instruction), the high-order 8 bits of the 16-bit I/O address must be 0.

The conventional I/O instruction (OUT (m),A/ IN A, (m) / OUTI /INI etc.) place the contents of a CPU register on the high-order 8 bits of the address bus, and may be difficult to use for accessing on chip I/O registers. For more efficient on-chip I/O register access, the AB181E-20 has additional instructions above the Z180 which force the high-order 8 bits of the I/O address to 0. These instructions are IN0, OUT0, OTIMR, OTDMR and TSTIO (See Appendicies at the back of this book).

When writing to an internal I/O register, the same I/O write occurs on the external bus. However, the duplicate external I/O write cycle will exhibit internal I/O write cycle timing. For example, the WAIT* input and programmable wait state generator are ignored. This will be the same for internal I/O read cycles - however, the external read data is ignored by the AB181E-20.

It is advised that external I/O addresses should be chosen to avoid overlap with internal I/O addresses. This avoids duplicate I/O accesses.

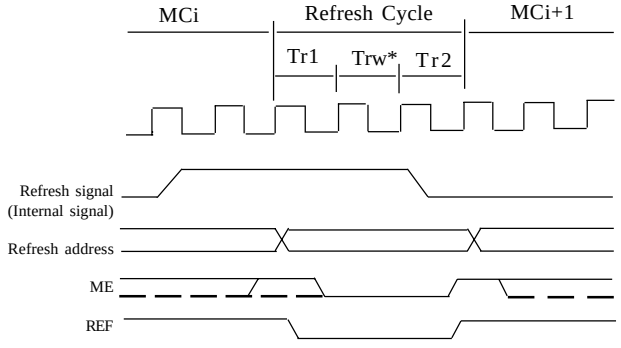
Dynamic RAM Refresh Control

The AB181E-20 incorporates a dynamic RAM refresh control circuit including 8 bit refresh address generation and programmable refresh timing. This circuit generates asynchronous refresh cycles inserted at the programmable interval independent of CPU program execution. For systems which don't use dynamic RAM, the refresh function can be disabled.

When the internal refresh controller determines that a refresh cycle should occur, the current cycle is inserted by placing the refresh address on A0-A7 and the REF* output is driven LOW.

By programming the REFW (Refresh Wait) bit in RCR (Refresh Control Register), the refresh cycles can be set for a clock cycle of either two or three. The external WAIT* input and the internal wait state generator are not effective during refresh.

Fig.6 shows the timing of a refresh cycle with a refresh wait (Trw) cycle.



Note: *If three cycles are specified, Trw is inserted. Otherwise, Trw is not inserted.
MC: Machine Cycle

Figure 6 Refresh Timing

Refresh Control Register (RCR)
RCR sets the interval and length of refresh cycles and enables or disables the refresh function.

bit 7	6	5	4	3	2	1	0
REFE	REFW	-	-	-	-	CYC1	CYC0
R/W		R/W				R/W	R/W

REFE: Refresh Enable (bit 7)
REFE = 0 disables the refresh controller while REFE = 1 enables refresh cycle insertion. REFE is set = 1 during RESET.

REFW: Refresh Wait (bit 6)
REFW = 0 causes the refresh cycle to be two clocks in duration. REFW = 1 causes the refresh cycle to be three clocks in duration by adding a refresh wait cycle (Trw). REFW is set = 1 during RESET.

CYC1,0: Cycle Interval (bit 1,0)
These two bits specify the interval between refresh cycles.

CYC1	CYC0	Interval (Cycles of PHY)	Interval ms (at PHY = 20MHz)
0	0	124	6.2
0	1	234	11.7
1	0	310	15.5
1	1	390	19.5

On Reset CYC1,0 are cleared to 0 giving the shortest refresh interval.

In the case of dynamic RAMs requiring 128 refresh cycles every 2 ms (or 256 cycles every 4ms), the required refresh interval is less than or equal to 15.625 ms.

Refresh Control and RESET
After RESET, based on the initialised value of RCR, refresh cycles will occur with an interval of 248 clock cycles and be 3 clock cycles in duration.

Dynamic RAM Refresh Operation Notes
Refresh cycle insertion is stopped when the CPU is in following states.

(a) During RESET
(b) When the bus is released in response to BUSREQ*
(c) During WAIT states

Refresh cycles are suppressed when the bus is released in response to BUSREQ*, but the refresh timer continues to operate, so the time at which the first refresh cycle occurs after the AB181E-20 re-acquires the bus depends on the refresh timer. There is no timing relationship with the bus exchange.

Note: each refresh bus cycle will use a refresh address that has increased by 1 from the previous refresh bus cycle (This is for each completed refresh cycle, not each refresh request).

Logical Address Spaces

The 64k bytes CPU logical address space is interpreted by the MMU as consisting of up to three separate logical address areas, Common Area 0, Bank Area and Common Area 1.

Fig 7 displays the logical memory configurations that are possible. The boundaries between the Common and Bank areas can be programmes with 4k bytes resolution.

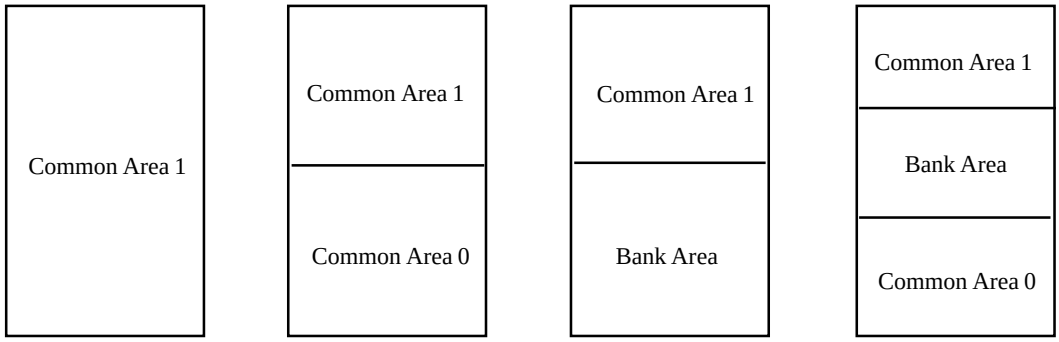


Figure 7 **Logical Address Mapping Examples**

Logical to Physical Address Translation

Fig 8 is an example in which the three logical address space portions are mapped into a 1M bytes physical address space. It can be seen that Common and Bank areas can overlap and that Common Area 1 and Bank Area can be relocated (on 4k bytes physical address boundaries). Common Area 0 (if it exists) is always based at physical address 0.

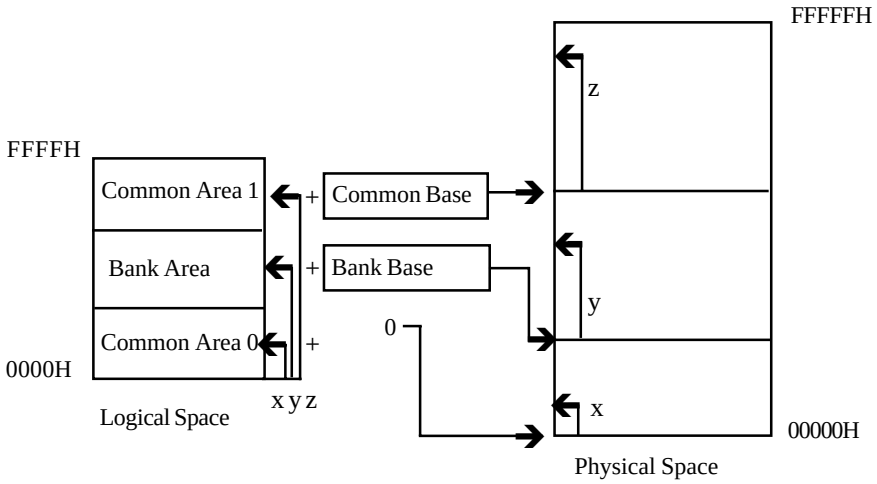


Figure 8 **Logical ➔ Physical Mapping Example**

Memory Management Unit (MMU)

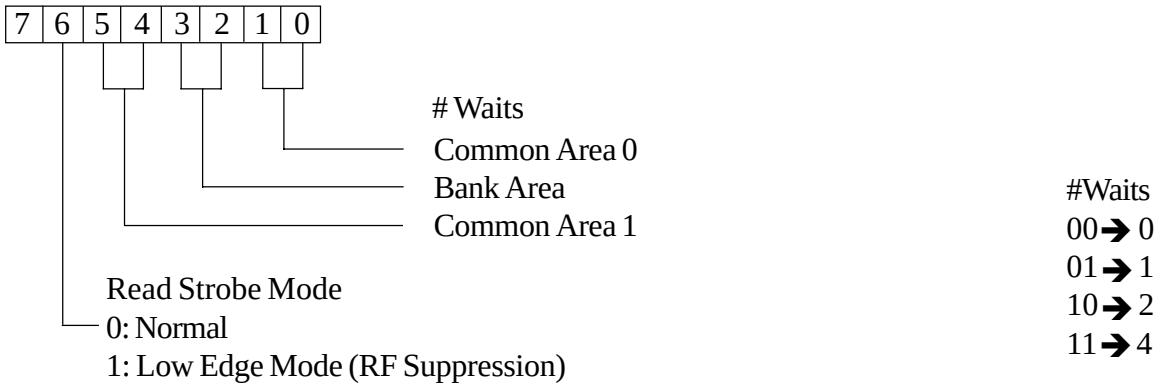
The AB181E-20 contains an on-chip MMU which performs the translation of the CPU 64k bytes (16-bit addresses- 0000H to FFFFH) logical memory address space into a 1M bytes (20-bit addresses- 00000H toFFFFFH) physical memory address space. Address translation is performed internally in parallel with other CPU operation.

Memory Wait State Control

Each logical memory address space can have an independantly assigned number of wait states. This is controlled through the IMWR register (2dh).

The internal memory map and register function is similar to the Z180, with the following additions.

Internal Memory Wait Control Register (IMWR : I/0 Address 2dH).



NB: When using the IMWR to set different numbers of wait states for each bank it is required that the following restriction is observed:

CBR ³ BBR

I/0 Wait State Control

The number of wait states inserted by the bus controller during I/0 cycles can be set through the DMA/ WAIT Control Register (DCNTL).

DMA/WAIT Control Register (DCNTL : I/O Address = 32H)

bit	7	6	5	4	3	2	1	0
	——	——	IWII	IWI0	DMS1	DMS0	DIM1	DIM0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The IWII and IWIO bits control the number of wait states automatically inserted into I/0 cycles by the bus controller.

IWII	IWIO	# of Waits
0	0	0
0	1	2
1	0	4
1	1	7

On Reset IWII and IWI0 are set to 11 giving 7 wait states on all I/0 cycles.

For details on the DMSn and DIMn bits refer to the DMA Controller section.

MMU Block Diagram

Fig 20 shows the MMU block diagram. The MMU translates internal 16-bit logical addresses to external 20-bit physical addresses.

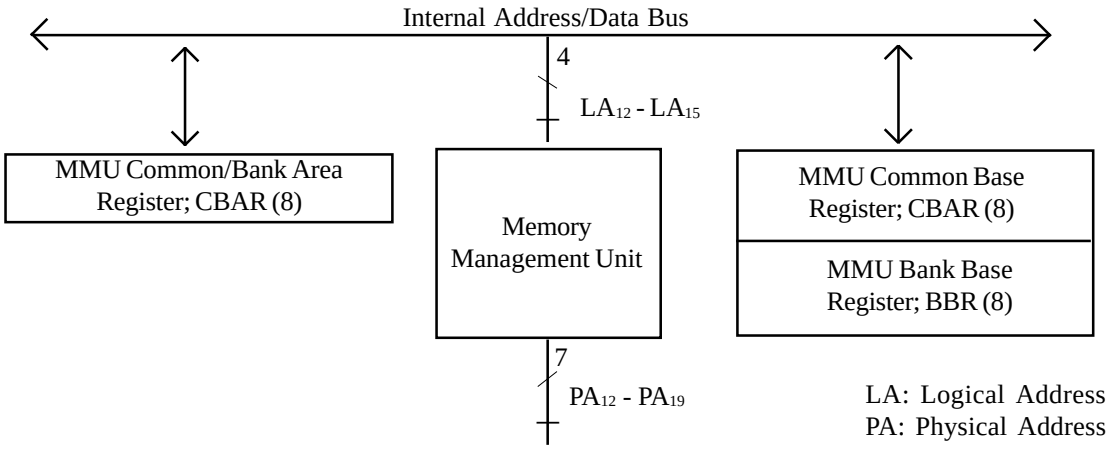


Figure 20 **MMU Block Diagram**

Whether address translation takes place depends on the type of CPU cycle as follows:

Memory Cycles
Address Translation occurs for all memory access cycles including instruction and operational fetchs, memory data reads and write, hardware interrupt vector fetch and software interrupt restarts.

I/O Cycles
The MMU is logically bypassed for I/O cycles. The 16-bit logical I/O address space corresponds directly with the 16-bit physical I/O address space. The four high order bits (A16-A19) of the physical address are always 0 during I/O cycles.

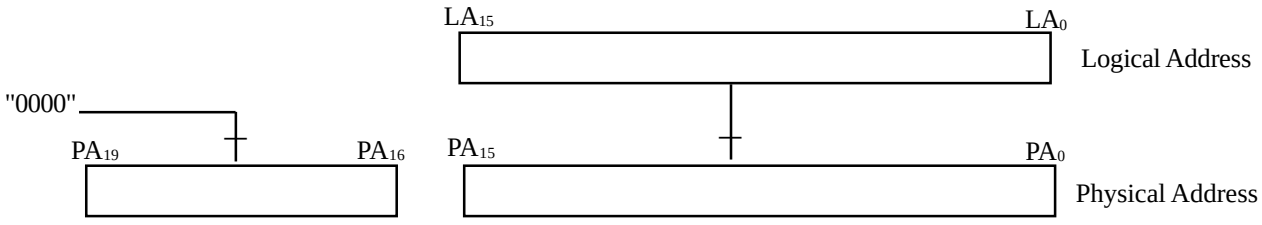


Figure 21 **I/O Address Translation**

DMA Cycles
When the AB181E-20 on-chip DMAC is using the external bus, the MMU is physically bypassed. The 20-bit source and destination registers in the DMAC are directly output on the physical address bus (AO-A19).

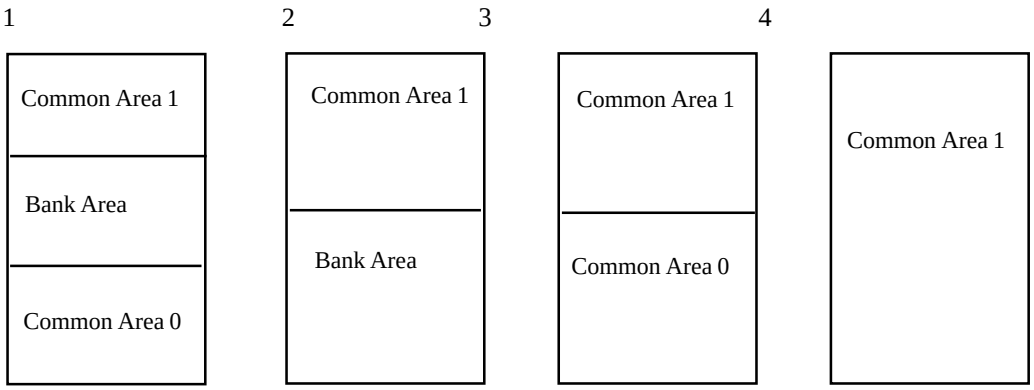
MMU Register

Three MMU registers are used to program a specific configuration of logical and physical memory.
(1) MMU Common/Bank Area Register (CBAR)
(2) MMY Common Base Register (CBR)
(3) MMU Bank Base Register (BBR)

CBAR is used to define the logical memory organisation. CBR and BBR are used to relocate logical areas within the 512k bytes physical address space. The resolution for setting boundaries within space and relocation within the physical space is 4k bytes.

The CAR fiels of CBAR determines the start address of Common Area 1 (Up-per Common) and by default, the end address of the Bank Area. The BAR field determines the start address of the Bank Area and by default, the end address of Common Area 0 (Lower Common).

The CA and BA field of CBAR may be programmed subject to the restriction that CA may never be less than BA Fig 22 and Fig 23 show examples of logical memory organisations associated with different values of CA and BA.



1	2	3	4
Common Area 1 Lower limit address	Common Area 1 Lower limit address	Common Area 1 Lower limit address	Common Area 1 Lower limit address
>	>	=	=
Bank Area Lower limit address	Bank Area Lower limit address	Bank Area Lower limit address	Bank Area Lower limit address
>	=	>	=
0000H	0000H	0000H	0000H

(RESET condition)

Figure 22 Logical Memory Organisation

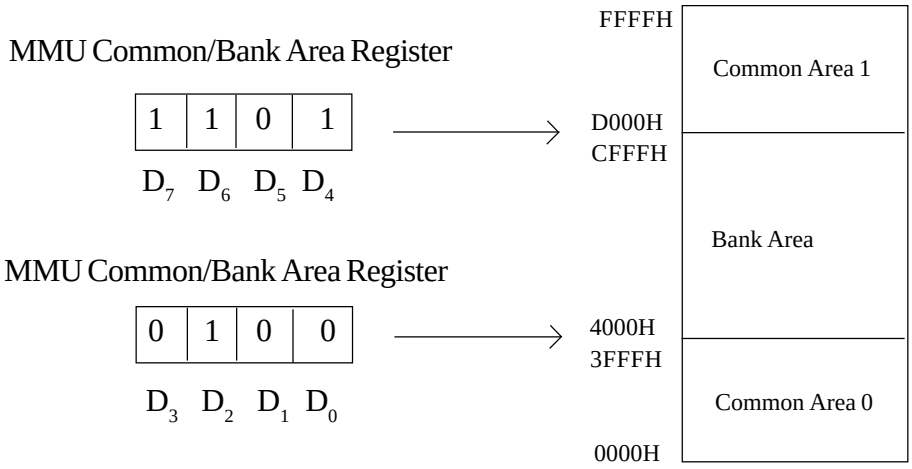


Figure 23 Logical Space Configuration (Example)

MMU Register Description

MMU Common/Bank Area Register (CBAR)

CBAR specifies boundaries within the AB181E-20 64k bytes logical address space for up to three areas, Common Area 0, Bank Area and Common Area 1.

MMU Common/Bank Area Register (CBAR : I/O Address = 3AH)

bit	7	6	5	4	3	2	1	0
	CA3	CA2	CA1	CA0	BA3	BA2	BA1	BA0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CA3-CA0: CA (bits 7-4)
CA specifies the start (low) address (on 4k bytes boundaries) for the Common Area 1. This also determines the last address of the Bank Area. All bits of CA are set to 1 during RESET.

BA3-BAO: BA (bits 3-0)
BA specifies the start (low) address (on 4k bytes boundaries) for the Bank Area. This also determines the last address of the Common Area 0. All bits of BA are reset to 0 during RESET.

MMU Common Base Register (CBR)

CBR specifies the base address (on 4k boundaries) used to generate 19-bit physical address for Common Area 1 accesses. All bits of CBR are reset to 0 during RESET.

MMU Common/Bank Area Register (CBAR : I/O Address = 38H)

bit	7	6	5	4	3	2	1	0
	CB7	CB6	CB5	CB4	CB3	CB2	CB1	CB0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MMU Bank Base Register (BBR)

CBR specifies the base address (on 4k boundaries) used to generate 19-bit physical address for Common Area 1 accesses. All bits of CBR are reset to 0 during RESET.

MMU Bank Base Register (BBR : I/O Address = 39H)

bit	7	6	5	4	3	2	1	0
	BB7	BB6	BB5	BB4	BB3	BB2	BB1	BB0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Physical Address Translation

Fig 24 shows the way in which physical addresses are generated based on the contents of CBAR, CBR and BBR. MMU comparators classify an access by logical area as defined by CBAR. Depending on which of the three potential logical areas (Common Area 1, Bank Area or Common Area 0) is being accessed, the appropriate 7-bit base address is added to the high-order 4 bits of the logical address, yeilding a 20-bit physical address. CBR is associated with Common Area 1 accesses. Common Area 0 accesses use an internal base register (non-accessible) which contains 0. Thus, Common Area 0, if defined, is always based at physical address 0.

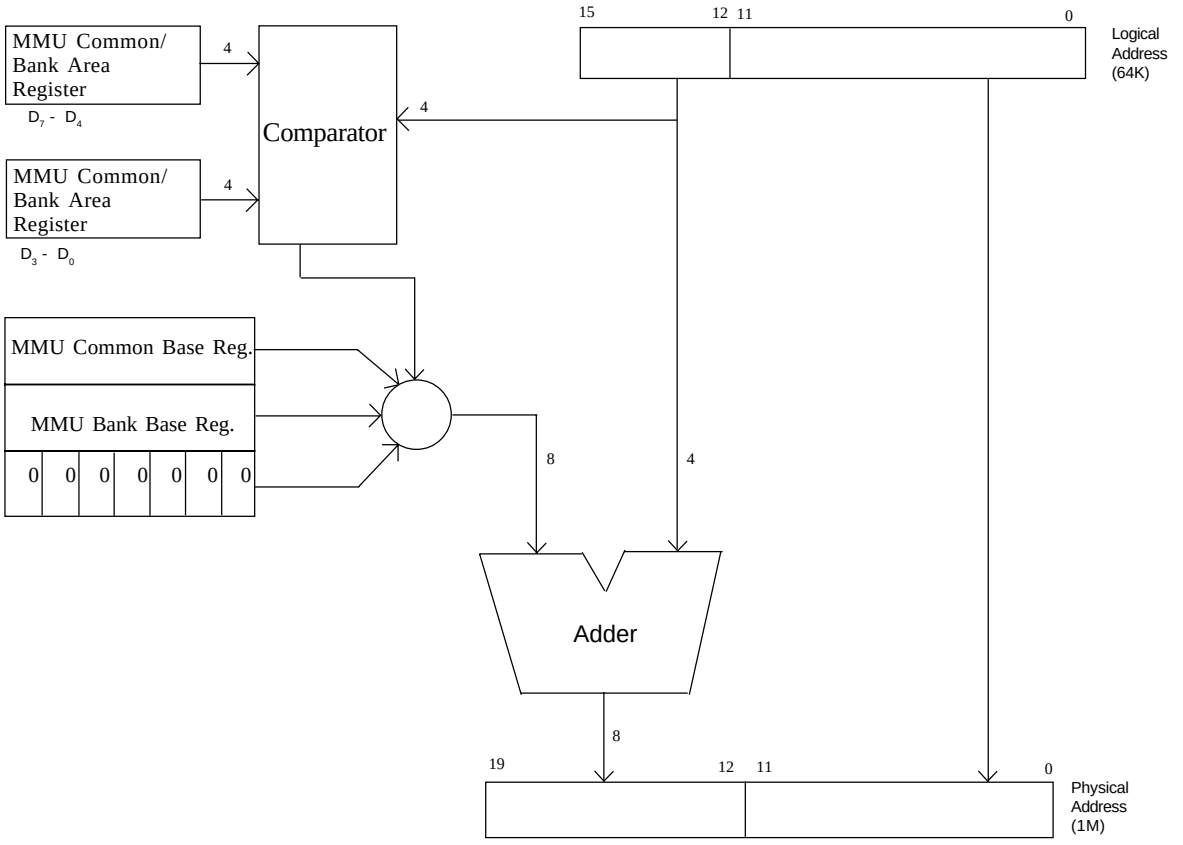


Figure 24 **Physical Address Generation**

MMU and Reset

During REST, all bits of the CA field of CBAR are set to 1. All bits of the BA field of CBAR, CBR and BBR are reset to 0. The logical 64k bytes address space corresponds directly with the first 64k bytes (0000H to FFFFH) of the 1M bytes (00000H to FFFFFH) physical address space, so after RESET the AB181E-20 will begin execution at logical and physical address 0.

MMU Register Access Timing

When data is written into CBAR, CBR or BBR, the value will be effective from the cycle immediately following the I/O write cycle which updates these registers.

Note: care must be taken during MMU programming to insure that CPU program execution is not disrupted. The next cycle following MMU register programming will normally be an op-code fetch from the newly translated address. One way of ensuring this is to localise all MMU programming routines in a Common Area that is always enabled.

Interrupt Control

The AB181E-20 CPU has thirteen interrupt sources, four external and nine internal, with fixed priority.

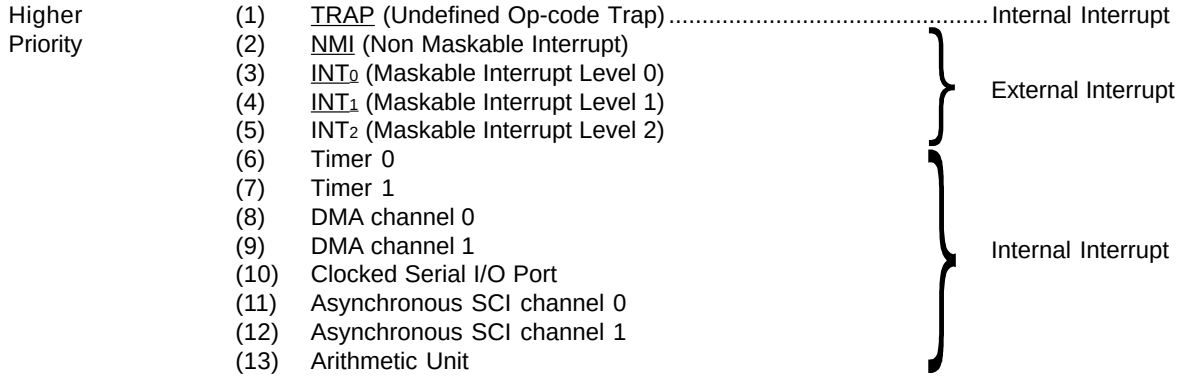


Figure 9 **Interrupt Sources**

This section explains the CPU registers associated with interrupt processing, the TRAP interrupt, interrupt response modes and the external interrupts. For full information on internal interrupt generation other than TRAP see the sections ASCI, DMA, PRT and CSI/O as required.

Interrupt Control Registers and Flags

The AB181E-20 contains three registers and two flags which are associated with interrupt processing.

	Function	Name	Access Method
(1)	Interrupt Vector High	I	LD A, I and LD I, A instructions
(2)	Interrupt Vector Low	IL	I/O instruction (addr=33H)
(3)	Interrupt/Trap Control	ITC	I/O instruction (addrs=34H)
(4)	Interrupt Enable Flag 1,2	IEF1,2	EI and DI LD A, I LD A, R instructions

Interrupt Vector Register (I)

External interrupts INT1, INT2 and all internal interrupts (except TRAP) use a programmable vectored technique (similar to mode 2) to determine the address at which interrupt processing starts. In response to the interrupt a 16-bit address is generated. This address accesses a vector table in memory to obtain the address at which execution restarts.

INT Ø generates a jump to an ISR at 0038h

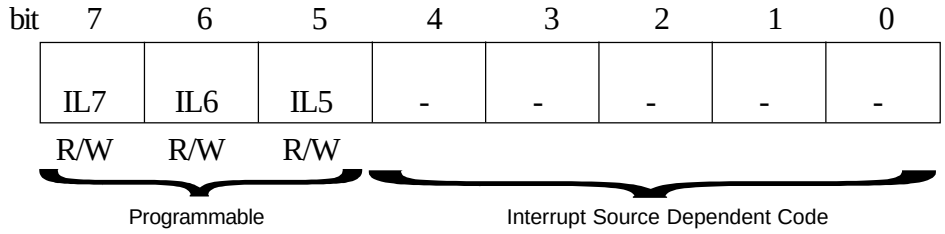
The methods for generation of the least significant byte of the table address are different, but all vectors use the contents of I as the most significant byte of the table address. By programming the contents of I, vector tables can be relocated on 256 bytes boundaries throughout the 64k logical address space.

Note: that I is read/written with the LD A, I and LD I, A instructions rather than I/O (IN, OUT) instructions.

I is initialized to 0 during RESET.

Interrupt Vector Low Register (IL)

Interrupt Vector Low Register (IL: I/O Address = 33H)

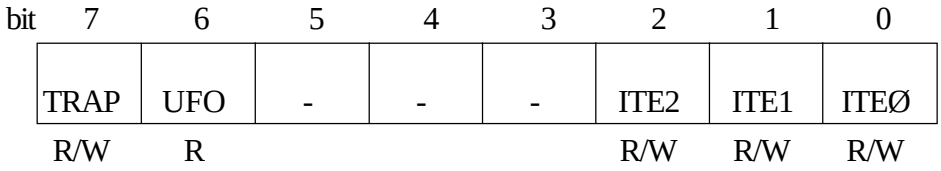


This register determines the most significant three bits of the low order byte of the interrupt vector table address for external interrupt INT1 and INT2 and all internal interrupts (except TRAP). The five least significant bits are fixed for each specific interrupt source. By programming IL the vector table can be relocated on 32 bytes boundaries.

IL is initialised to 0 during RESET.

INT/TRAP Control Register (ITC)

INT/TRAP Control Register (ITC : I/O Address = 34H)



ITC is used to handle TRAP interrupts and to enable or disable the external maskable interrupt inputs INT0*, INT1* and INT2*.

TRAP (bit 7)
This bit is set to 1 when an undefined up-code is fetched. TRAP can be reset under program control by writing with 0, however it cannot be written with 1 under program control. TRAP is reset to 0 during RESET.

UFO: Undefined Feth Object (bit 6)
When a TRAP interrupt occurs (TRAP bit set to 1), the contents of UFO allow determination of the starting address of the undefined instruction. This is necessary since the TRAP may occur on either the third byte of the op-code. UFO allows the stacked PC value (stacked in response to TRAP) to be correctly adjusted. If UFO = 0, the first op-code should be interpreted as the stacked PC - 1. If UFO = 1, the first op-code address is stacked PC-2. UFO is read-only.

ITE2, 1,0: Interrupt Enable 2, 1,0 (bits 2-0)
ITE2, ITE1, and ITE0 enable and disable the external interrupt inputs INT2*, INT1* and INT0* respectively. If reset to 0, the interrupt is masked. During RESET, ITE0 is initialised to 1 while ITE1 and ITE2 are initialised to 0.

Interrupt Enable Flag 1,2 (IEF1,2)
IEF1 controls the overall enabling and disabling of all internal and external maskable interrupts (i.e. all interrupts except NMI and TRAP).

If IEF1 = 0, all maskable interrupts are disabled. IEF1 can be reset to 0 by the DI (Disable Interrupts) instruction and set to 1 by the EI (Enable Interrupts) instruction.

The purpose of IEF2 is to correctly manage the occurance of NMI. During NMI the prior interrupt reception state is saved and all maskable interrupts are automatically disabled (IEF1 copied to IEF2 and then IEF1 cleared to 0). At the end of the NMI interrupt service routine, execution of the RETN (Return from Non-maskable Interrupt) will automaticaly restore the interrupt receiving state (by copying IEF2 to IEF1) prior to the occurance of NMI.

IEF2 state can be reflected in the P/V bit of the CPU Status register by execution of the (a) LD A, I or (b) LD A, R instructions.

Table 10 shows the state of IEF1 and IEF2.

CPU Operation	IEF1	IEF2	REMARKS
RESET	0	0	Inhibits the interrupt except NMI and TRAP
NMI	0	IEF1	Copies the contents of IEF1 to IEF2
RETN	IEF2	not affected	Returns from the NMI service routine
Interrupt except NMI and TRAP	0	0	Inhibits the interrupt
RET1	not affected	not affected	
TRAP	not affected	not affected	
EI	1	1	
DI	0	0	
LD A, I	not affected	not affected	Transfers the contents of IEF2 to P/V flag
LD A, R	not affected	not affected	Transfers the contents of IEF2 to P/V flag

Table 10 State of IEF1 and IEF2

TRAP Interrupt

The AB181E-20 creates a non-maskableTRAP interrupt when an undefined op-code fetch occurs (This is not affected by the of IEF1). This feature can be used to increase software reliability, implement an 'extended' instruction set, or both. TRAP may occur during op-code fetch cycles.

When a TRAP interrupt occurs the AB181E-20 operates as follows;
(1) The TRAP bit in the Interrupt TRAP/Control (ITC) register is set to 1.

(2) The current PC (Program Counter) value, reflecting the location after the undefined op-code, is saved on the stack.

(3) The AB181E-20 vectors to logical address 0. If logical address 0 is mapped to physical address 0, the vector is the same as for RESET. Testing the TRAP bit in ITC will reveal whether the restart at physical address 0 was caused y RESET or TRAP. The state of the UFO (Undefined Fetch Object) bit in ITC allows TRAP handling software to correctly adjust the stacked PC depending on whether the second or third byte of the op-code generated the TRAP. If UFO = 0, the starting address of the invalid instruction is equal to the stacked PC-2. If UFO = 1, the starting address of the invalid instruction is equal to the stacked PC-4. Fig 11 shows TRAP Timing.

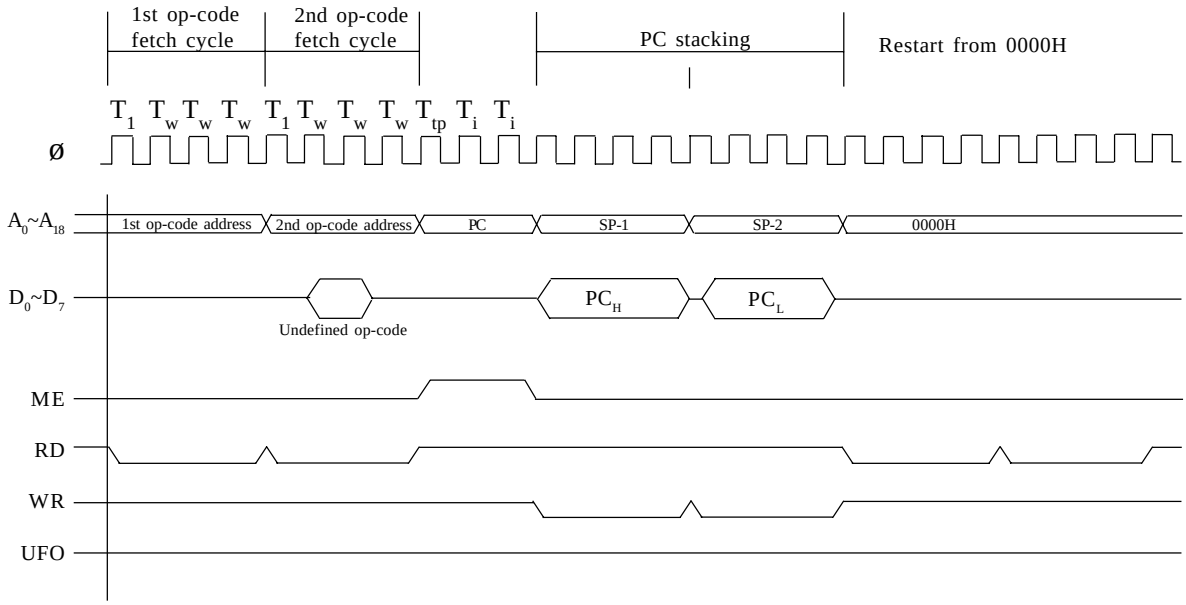


Figure 11 **TRAP - 2nd Op-code Undefined**

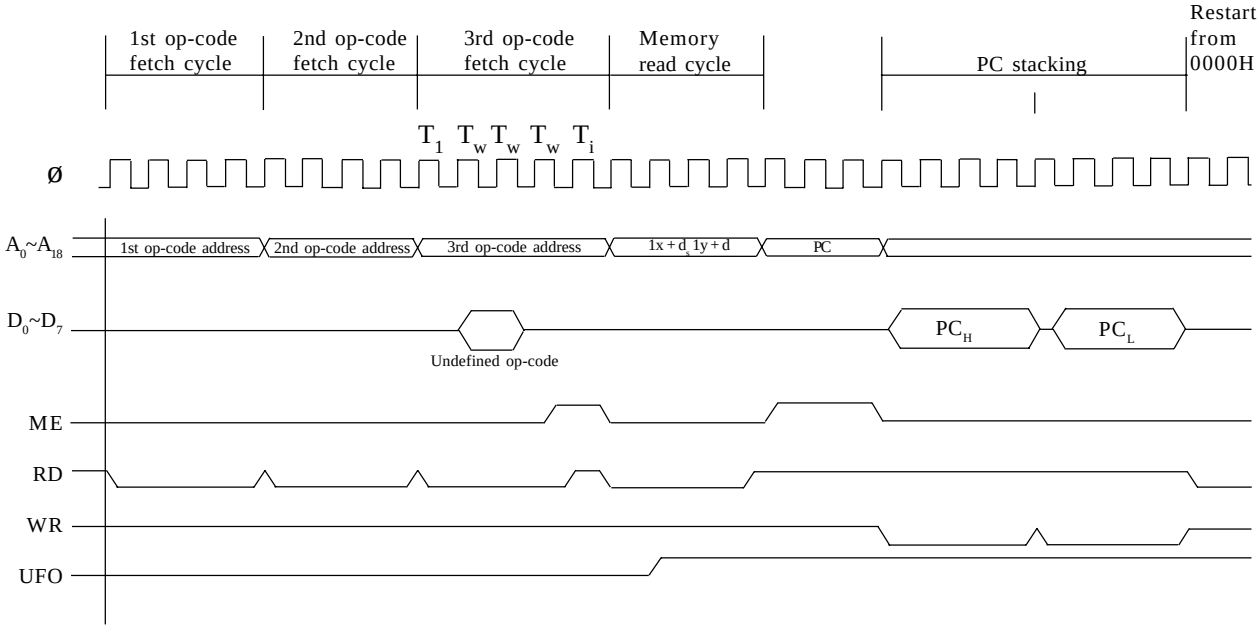


Figure 12 **TRAP - 3rd Op-code Undefined**

External Interrupts

The AB181E-20 has four external hardware interrupt inputs.

(1) NMI - Non-Maskable Interrupt
(2) INT0 - Maskable Interrupt Level 0
(3) INT1 - Maskable Interrupt Level 1
(4) INT2 - Maskable Interrupt Level 2

NMI, INT0, INT1 and INT2 have fixed interrupt response modes.

NMI - Non-Maskable Interrupt

The NMI* interrupt input is edge sensitive and cannot be masked by software. When NMI* is detected, the AB181E-20 operates as follows:

(1) DMAC operation is suspended by the clearing of the DME (DMA Main Enable) bit in DCNTL.
(2) The PC is pushed onto the stack.
(3) The contents of IEF1 are copied to IEF2. This saves the interrupt reception state that existed prior to NMI.
(4) IEF1 is cleared to 0. This disables all external and internal maskable interrupts except NMI and TRAP).
(5) Execution commences at logical address 66H.
The last instruction of an NMI service routine should be RETN (Return from Non-maskable Interrupt). This restores the stacked PC, allowing the interrupted program to continue, and RETN causes IEF2 to be copied to IEF1, restoring the interrupt reception state that existed prior to the NMI.

Note: since NMI can be accepted during AB181E-20 on-chip DMAC operation, it can be used to externally interrupt DMA transfer. The NMI service routine can reactivate or abort the DMA operation as required by the application.

Special care must be taken to insure that interrupt inputs do not over run the NMI service routine. Unlimited NMI* inputs without a corresponding number of RETN instructions will eventually cause stack overflow.

Fig 13 shows the use of NMI and RETN and Fig. 14 details NMI response timing. The NMI response sequence is activated if the internally latched edge sensitive NMI* input is detected at the rising edge of Ø.

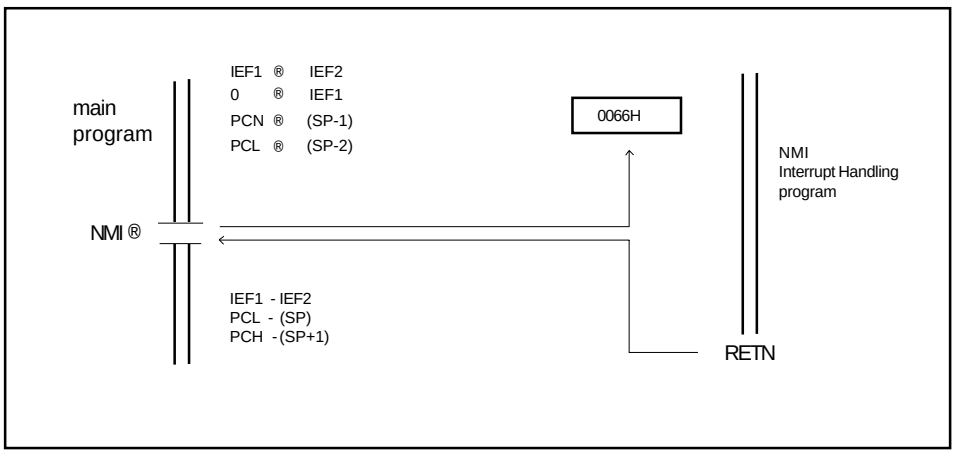


Figure 13 **NMI Sequence**

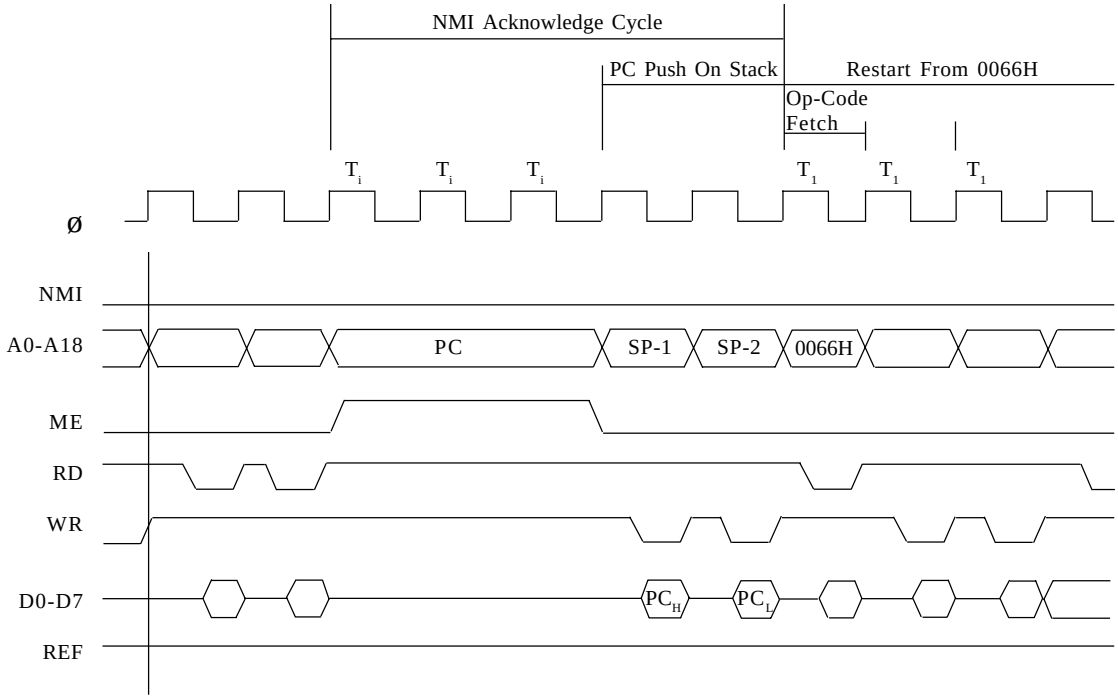


Figure 14 **NMI Timing**

INT0 - Maskable Interrupt Level 0

The next highest priority external interrupt after NMI is INT0. The interrupt is masked if either the IEF1 flag or the ITE0 (Interrupt Enable 0) bit in ITC are reset to 0. Note that after RESET the state is as follows:

(1) IEF1 is 0, so INT0 is masked.

(2) ITE0 is 1, so INT0 is enabled by execution of the EI (Enable Interrupts) instruction.

INT0

When INT0* is received, the PC is stacked and instruction execution restarts at logical address 38H. Both IEF1 and IEF2 flags are reset to 0, disabling all maskable interrupts. The interrupt service routine normally terminates with the EI (Enable Interrupts) instruction followed by the RETI (Return fro Interrupt) instruction, so that the interrupts are re-enabled. Fig 15 shows the use of INT0 and RETI. Fig 16 shows INT0 Response Timing.

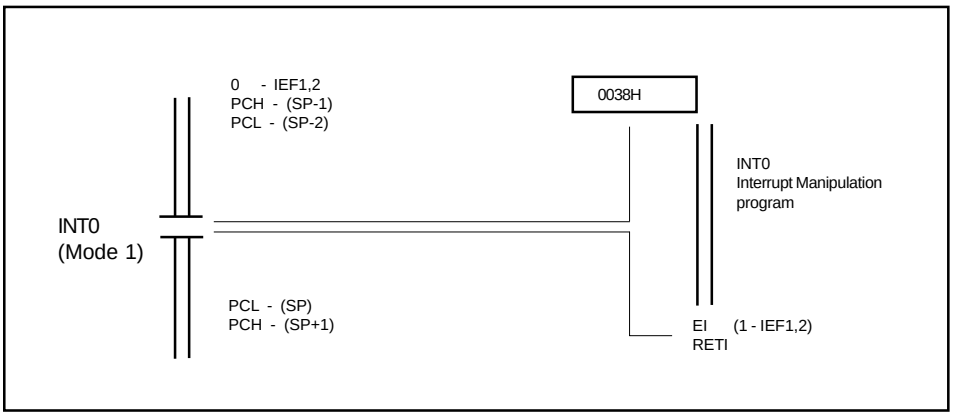


Figure 15 **INT0 Interrupt Sequence**

Note that TRAP interrupt will occur if an invalid instruction is fetched during an interrupt acknowledge cycle.

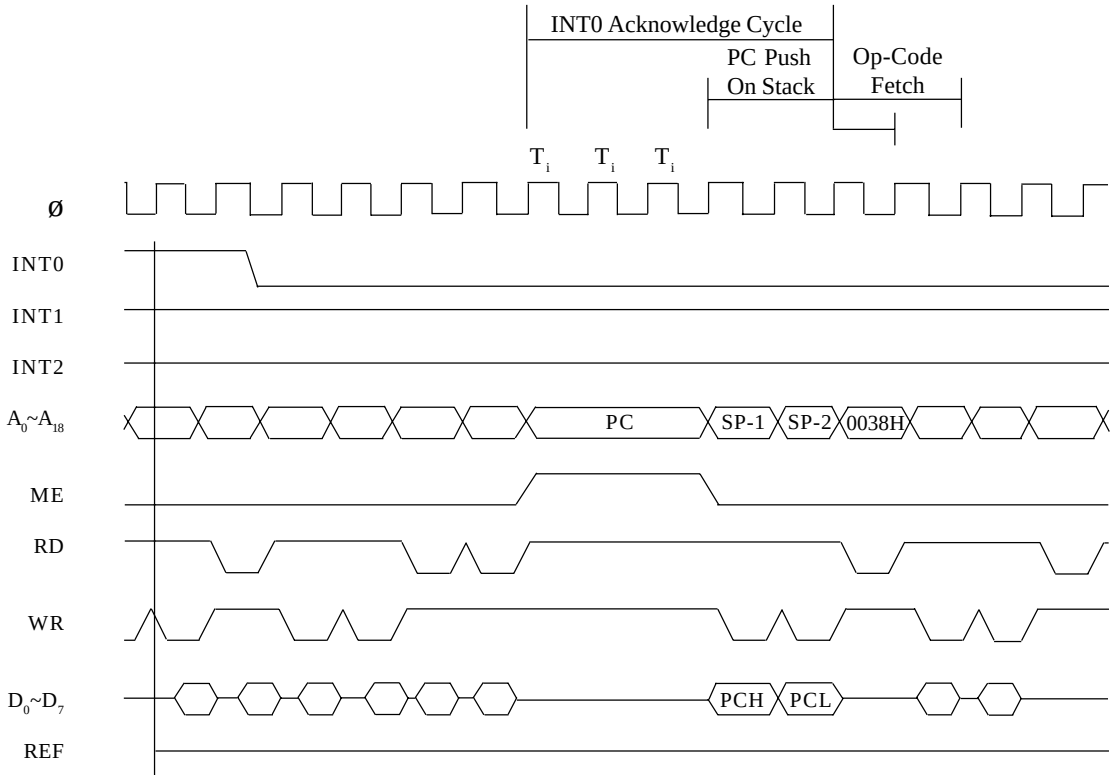


Figure 16 **INT0 Timing**

INT1, INT2

The operation of external interrupts INT1 and INT2 is a vector mode. INT1 and INT2 generate the low-order byte of vector table address using the IL (interrupt Vector Low) register and this is also the interrupt response used for all internal interrupts (except TRAP).

The low-order byte of vector table address is comprised of the most signficant three bits of the software programmable IL register while the least significant five bits are a unique fixed value for each interrupt (INT1, INT2 and internal) source. This is shown in Fig 17

INT1* and INT2* are globally masked by IEF1 = 0. Each one is also individually maskable by respectively clearing the ITE1 and ITE2 (bits 1, 2) of the ITC register to 0.

During RESET, IEF1, ITE1 and ITE2 bits are reset = 0.

Internal Interrupts

Internal interrupts (except TRAP) use the same vectored response mode as INT1 and INT2. Internal interrupts are globally masked by IEF = 0. Individual internal interrupts are enabled/disabled by programming each individual peripheral (ASCI, DMAC, PRT, FPAU, CSI/O) control register. See Table 18 for a summary of the lower vector of INT1, INT2 and internal interrupts.

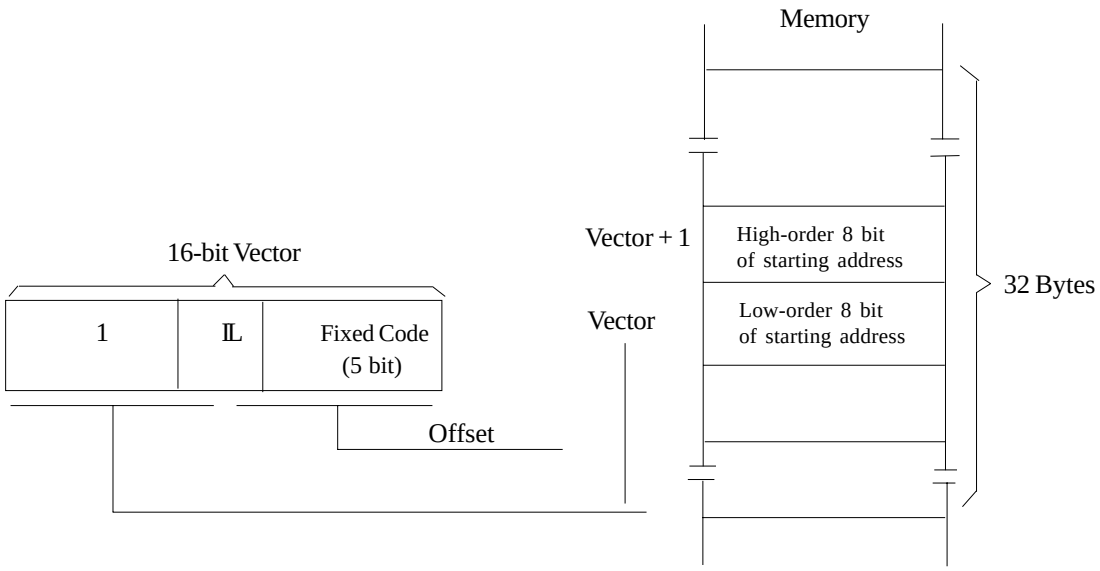


Figure 17 INT1, INT2 and Internal Interrupts Vector Acquisition

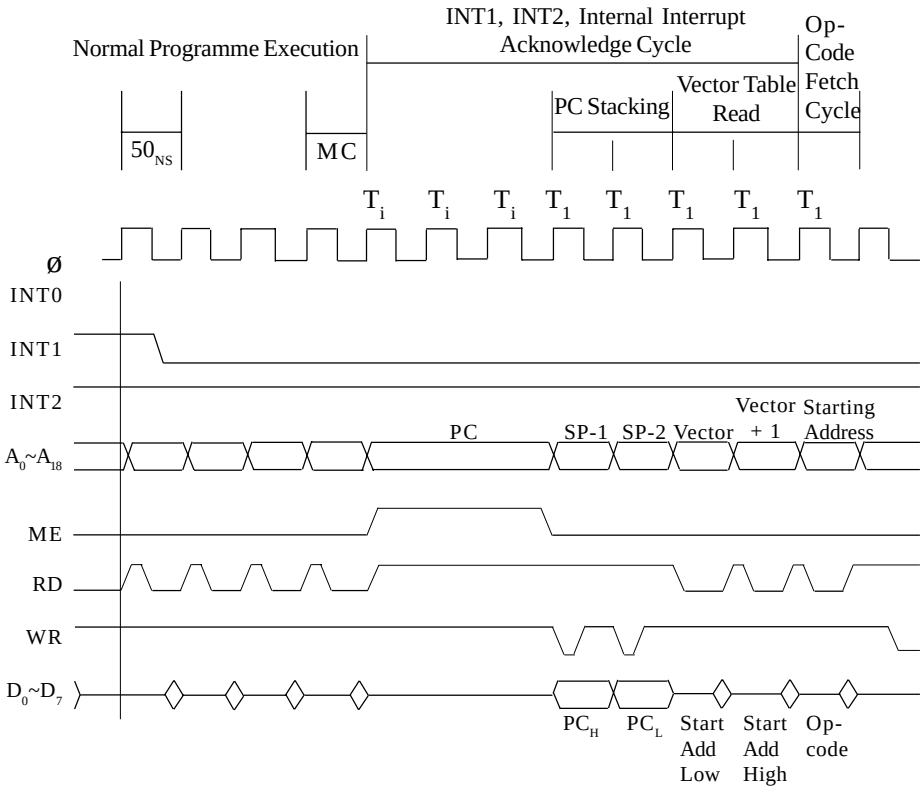
Interrupt Source	Priority	IL			Fixed Code				
		b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
INT1	Highest	*	*	*	0	0	0	0	0
INT2		*	*	*	0	0	0	1	0
Timer channel 0		*	*	*	0	0	1	0	0
Timer channel 1		*	*	*	0	0	1	1	0
DMA channel 0		*	*	*	0	1	0	0	0
DMA channel 1		*	*	*	0	1	0	1	0
CSI/O		*	*	*	0	1	1	0	0
ASCI channel 0		*	*	*	0	1	1	1	0
ASCI channel 1		*	*	*	1	0	0	0	0
Arithmetic Unit	Lowest	*	*	*	1	0	0	1	0

*Programmable

Table 18 Interrupt Source and Lower Vector

Interrupt Acknowledge Cycle Timing

Fig. 19 shows interrupt acknowledge cycle timing for internal interrupts, INT1 and INT2.



MC: Machine Cycle

Figure 19 INT1, INT2 and Internal Interrupts Timing

Interrupt Sources and RESET

I Register

All bits reset to 0.

1 = 0 locates the vector tables starting at logical address 0, so vectored interrupts (INT0 , INT1, INT2 and internal interrupts) will overlap with fixed restart interrupts like RESET (0), NMI (66H), INT0 Mode 1 (38H) and RST (00H - 38H). The vector table(s) can be built elsewhere in memory and located on 256 bytes boundaries by reprogramming I with the LD I, A instruction.

IL Register

Bits b7, b6 and b5 are reset to 0.

The IL register can be programmed to locate the vector for INT1, INT2 and internal interrupts on 32 bytes sub-boundaries within the 256 bytes area spcified by 1.

IEF1, IEF2 Flags

Reset to 0.

This disables all Interrupts other than NMI and TRAP.

ITC Register

ITE0 set to 1. ITE1, ITE2 reset to 0.

INT0* and INT2* requires that the ITE1 and ITE2 bits be respectively set = 1 by writing to ITC.

I/O Control Registers

Interrupt enable bits reset to 0.

All AB181E-20 on-chip peripheral interrupts are disabled and can be individually enabled by writing to each I/O control register interrupt enable bit.

Asynchronous Serial Communication Interface (ASCI)

The AB181E-20 on-chip ASCI contains two independent full duplex channels. The flexibility of the ASCI allows direct communication with a wide variety of standard UARTs (Universal Asynchronous Receiver Transmitter) . The key functions for ASCI are shown below. Each channel is independently programmable.

- Full duplex communication
- 7- or 8-bit data length
- 1 or 2 stop bits
- Odd, even, no parity
- Parity, overrun, framing error detection
- Programmable baud rate generator, /16 and /64 modes
- Speed to 125k bits per second (CPU fc =20 MHz)
- Modem control signals - Channel 0 has DCD0*, CTS0* and RTS0* Channel 1 has CTS1*
- Programmable interrupt condition enable and disable
- Operation with on-chip DMAC
- Flexible prescaler allows standard baud rates with any crystal frequency

ASCI Block Diagram

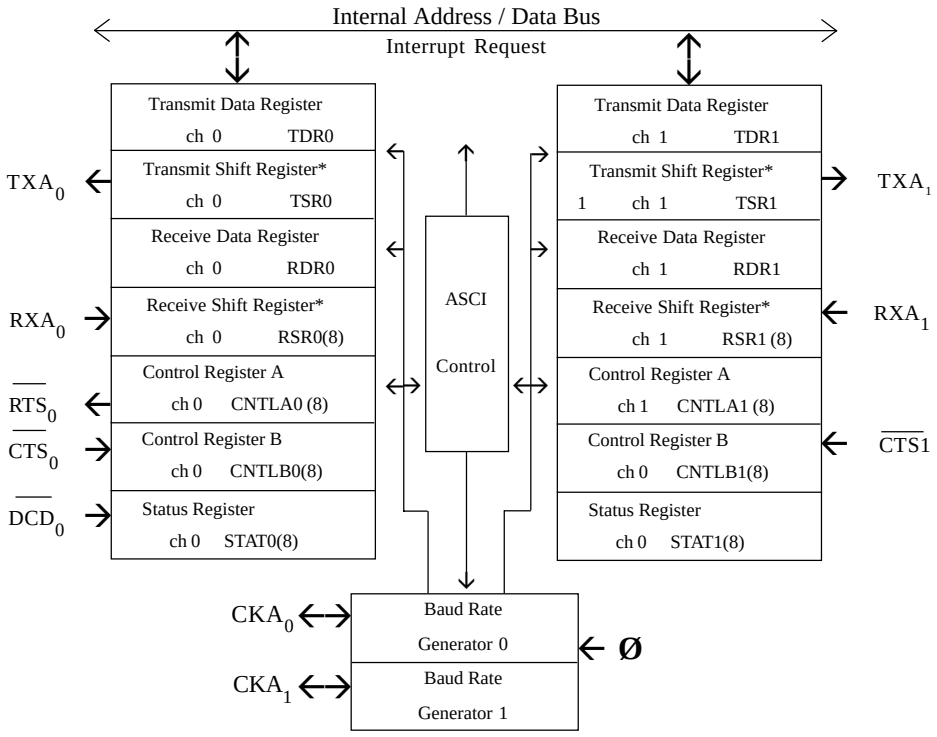


Figure 25 **ASCI Block Diagram**

ASCI Register Description

Transmit Shift Register 0, 1 (TSRO, 1)

When the Transmit Shift Register receives data from the Transmit Data Register (TDR) the data is shifted out of the TXA pin. When transmission is completed, the next byte (if available) is automatically loaded from TDR into TSR and the next transmission starts. The TSR outputs a continuous High level If no data is available for transmission. This register is not program accessible.

Transmit Data Register 0, 1 (TDR0,1: I/O Address = 06H, 07H)

Data written to the Transmit Data Register is transferred to the TSR when the TSR is empty. Data can be written to while TSR is shifting out the previous byte of data, making the ASCI transmitter double buffered.

Receive Shift Register 0, 1 (RSRO,1)

This register receives data shifted in on the RXA pin. When full, data is automatically transferred to the Receive Data Register (RDR) if it is empty. If RSR is not empty when the next incoming data byte is shifted in, an overrun error occurs. This register is not program accessible.

Receive Data Register 0, 1 (RDRO, 1:I/O Address = 08H, 09H)

When complete incoming data byte is assembled in RSR, it is automatically transferred to the RDR if RDR is empty. The next incoming data byte can be shifted into RSR while RDR contains the previous received data byte, so the ASCI receiver is double buffered.

ASCI Status Register 0, 1 (STAT0, 1)

Each channel status register allows interrogation of ASCI communication, error and modem control signal status as well as enabling and disabling of ASCI interrupts.

ASCI Status Register 0 (STAT0 : I/O Address = 04H)

bit	7	6	5	4	3	2	1	0
	RDRF	OVRN	PE	FE	RIE	DCD	TDRE	TIE
	R	R	R	R	R/W	R	R	R/W

ASCI Status Register 1 (STAT1 : I/O Address = 05H)

bit	7	6	5	4	3	2	1	0
	RDRF	OVRN	PE	FE	RIE	CTSIE	TDRE	TIE
	R	R	R	R	R/W	R	R	R/W

RDRF : Receive Data Register Full (bit 7)

RDRF is set = 1 when an incoming data byte is loaded into RDR. If a framing or parity error occurs, RDRF is still set and the received data (which generated the error) is still loaded into RDR. RDRF is cleared = 0 by reading RDR, when the DCD* input is HIGH, in IOSTOP mode and during RESET.

OVRN: Overrun Error (bit 6)
OVRN is set = 1 when RDR is full and RSR becomes full. OVRN is cleared = 0 when the EFR bit (Error Flag Reset) of CNTLA is written = 0, when DCD* is HIGH, in IOSTOP mode and during RESET.

PE: Parity Error (bit 5)
PE is set = 1 when a parity error is detected on an incoming data byte and ASCI parity detection is enable (the MOD1 bit of CNTLA set = 1). PE is cleared = 0 when the EFR bit (Error Flag Reset) of CNTLA is written = 0, when DCD* is HIGH, in IOSTOP mode and during RESET.

FE: Framing Error (bit 4)
If a receive data byte frame is delimited by an invalid stop bit (i.e. 0, should be 1), FE is set = 1. FE is cleared = 0 when the EFR bit (Error Flag Reset) of CNTLA is written = 0, when DCD* is HIGH, in IOSTOP mode and during RESET.

RIE: Receive Interrupt Enable (bit 3)
RIE should be set = 1 to enable ASCI receive interrupt requests. When RIE =1, if any of the flags RDRF, OVRN, PE FE become set = 1 and interrupt request is generated. For channel 0, an interrupt will also be generated by the transition of the external DCD0* input from LOW to HIGH. RIE is cleared = 0 during RESET.

DCD0*: Data Carrier Detect (bit 2 STAT0)
Channel 0 has an extenal DCD0* input pin. The DCD0* bit is set = 1 when the DCD0* input is high. It is cleared = 0 on the first read of STAT0 following the DCD0* input transition from high to low and during RESET. When DCD0* =1, receiver unit is reset and receiver operation is inhibited.

CTS1E: Channel 1 CTS* Enable (bit 2 STAT1)
Channel 1 has an external CTS1* input (pin 52) which is multiplexed with the receive data pin (RXS) for the CSI/O (Clocked Serial I/O Port). Setting CTS1E = 1 selects the CTs1* function and clearing CTS1E = 0 selects the RXS function.

TDRE: Transmit Data Register Empty (bit 1)
TDRE = 1 indicates that the TDR is empty and the next transmit data byte can be written to TDR. After the byte is written to TDR, TDRE is cleared = 0 until the ASCI transfers the bytes from the TDR to the TSR, at which time TDRE is again set = 1. TDRS is set = 1 in IOSTP mode and during RESET. When the external CTS* input is HIGH, TDRE is reset = 0.

TIE: Transmit Interrupt Enable (bit 0)
TIE should be set = 1 to enable ASCI transmit interrupt requests. If TIE = 1, an interrupt will be requested when TDRE = 1. TIE is cleared = 0 during RESET.

ASCI Control Register A0, 1 (CNTLA0, 1)
Each ASCI Channel Control Register A configures the major operating modes such as receiver/transmitter enable and disable, data format, and multiprocessor communication mode.

ASCI Control Register A 0 (CNTLA0 : Address = 00H)

bit 7	6	5	4	3	2	1	0
——	RE	TE	$\overline{RTS_O}$	EFR	MOD2	MOD1	MOD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ASCI Control Register A 1 (CNTLA0 : Address = 01H)

bit 7	6	5	4	3	2	1	0
——	RE	TE	CKA1D	EFR	MOD2	MOD1	MOD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

RE: Receiver Enable (bit 6)
When RE is set = 1, the ASCI receiver is enabled. When RE is reset = 0, the receiver is disabled and any receive operation in progress in interrupted, but the RDRF and error flags are not reset and the previous contents of RDRF and error flags are held. RE is cleared = 0 in IOSTOP mode and during RESET.

TE: Transmitter Enable (bit 5)
When TE is set = 1, the ASCI transmitter is enabled. When TE is reset = 0, the transmitter is disabled and any transmit operation in progress in interrupted. However, the TDRE flag is not reset and the previous contents of TDRE are held. TE is cleared = 0 in IOSTOP mode and during RESET.

RTSO* - Request to Send Channel 0 (bit 4 inCNTLA0)
When RTS0* is reset = 0, the RTS0* output pin will go low. When RTS0* is set = 1, the RTS0* output immediately goes high. RTS0* is set = 1 during RESET.

CKA1D: CKA1 Clock Disable (bit 4 in CNTLA1)
When CKA1D is set = 1, the multiplexed CKA1/TEND0* pin (pin 50) is used for the TEND0* function. When CKA1D = 0, the pin is used as CKA1, an external data clock input/output for channel 1. CKA1D ic cleared = 0 during RESET.

EFR: Error Flag Reset (bit 3)
When written = 0, the EFR function is selected to reset all error flags (OVRN, FE and PE) to 0. EFR is undefined during RESET.

MOD2, 1,0: ASCI Data Format Mode 2, 1, 0 (bits 2-0)
These bits program the ASCI data format as follows

MOD2		MOD1		MOD0	
=	0 → 7 bit data	=	0 → No Parity	=	0 → 1 stop bit
=	1 → 8 bit data	=	1 → Parity enabled	=	1 → 2 stop bits

The data formats available based on all combinations of MOD 2, MOD1 and MOD0 are shown as follows.

MOD2	MOD1	MOD0	Data Format
0	0	0	Start + 7 bit data + 1 stop
0	0	1	Start + 7 bit data + 2 stop
0	1	0	Start + 7 bit data + parity + 1 stop
0	1	1	Start + 7 bit data + parity + 2 stop
1	0	0	Start + 8 bit data + 1 stop
1	0	1	Start + 8 bit data + 2 stop
1	1	0	Start + 8 bit data + parity + 1 stop
1	1	1	Start + 8 bit data + parity + 2 stop

ASCI Control Register B0, 1 (CNTLB0, 1)
Each ASCI channel control register B configures multiprocessor mode, parity and baud rate selection.

ASCI Control Register B 0 (CNTLB0 : I/O Address = 02H)
ASCI Control Register B 1 (CNTLB1 : I?O Address = 03H)

CTS*: Clearto Send
When read, CTS* reflects the state of the external CTS* input. If the CTS* input pin is high, CTS* will be read as 1. When the CTS* input pin is high, the TDRE bit is inhibited (i.e. held at 0). For channel 1, the CTS1* input is multiplexed with RXS pin (Cloaked Serial Receive Data), so CTS* is only valid when read if the channel 1 CTS1E bit = 1 and the CTS1* input pin function is selected. The read data of CTS* is not affected by RESET.

bit	7	6	5	4	3	2	1	0
	——	——	CTS	PEO	DR	SS2	SS1	SS0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PEO: Parity Even Odd (bit 4)
PEO selects even or odd parity. PEO does not affect the enabling/disabling of parity (MOD1 bit of CNTLA). If PEO is cleared = 0, even parity is selected. If PEO is set = 1, odd parity is selected. PEO is cleared = 0 during RESET.

DR: Divide Ratio (bit 3)
DR specifies the divider used to obtain baud rate from the data sampling clock. If DR is reset = 0, divide by 16 is used while DR is set = 1, divide by 64 is used. DR is cleared = 0 during RESET.

SS2, 1, 0: Source/Speed Select 2, 1, 0 (bits 2-0)
Specify the data clock source (internal or external) and baud rate prescale factor.
SS2, SS1, SS0 are all set = 1 during RESET. Table 26 shows the divide ratio corresponding to SS2, SS1 and SS0.

SS2	SS1	SS0	Divide Ratio
0	0	0	2
0	0	1	4
0	1	0	8
0	1	1	16
1	0	0	32
1	0	1	64
1	1	0	128
1	1	1	external clock

Table 26 Divide Ratio

The external ASCI channel 0 data clock pins are multiplexed with DMA control lines (CKA0/DREQ0* and CKA1/TEND0*). During RESET, these pins are initialised as ASCI data clock input. If SS2, SS1 and SS0 are reprogrammed (any other value than SS2, SS1 SS0 = 1) these pins become ASCI data clock outputs. However, if DMAC channel 0 is configured to perform memory « I/O (and memory mapped I/O transfers the CKA0/DREQ0* pins revert to DMA control signals regardless of SS2, SS1, SS0 programming. Also, if the CKA1D bit in the CNTLA register is set = 1, then the CKA1/TEND0* reverts to the DMA Control output function regardless of SS2, SS1 and SS0 programming.

Final data clock rates are based on PRSCALE, DR, SS2, SS1, and SS0 clock (Ø)frequency.

ASCI Baud Rate Prescale Register - PRSCALE 18H

The value written to this location determines the division ratio of the prescaler for the ASCI baud rate generators. This prescaler drives ASCI0 and ASCI1.

The baud rate is determined by

$$\frac{\text{(XTAL frequency *4)}}{\text{PRSCALE} * \begin{bmatrix}\text{16 if DR=0}\\ \text{64 if DR=1}\end{bmatrix} * \begin{bmatrix}\text{divide ratio set by SS}\\ \text{see Table 26}\end{bmatrix}}$$

Zero is an invalid value, unpredictable behaviour may result. Reading this register will return a number from a free-running counter.

Modem Control Signals

ASCI channel 0 has CTS0*, DCD0* and RTS0* external modem control signals. ASCI channel 1 has aCTS1* modem signal which is multiplexed with RXS pin (Clocked Serial Receive Data).

CTS0*: Clear to Send 0 (input)

The CTS0* input allows external control (start/stop) of ASCI channel 0 transmit operation. When CTS0* is high, channel 0 TDRE bit is held at 0 regardless of whether the TDR0 (Transmit Data Register) is full or empty. When CTS0* is low, TDRE reflects the state of TDR0. The actual transmit operation is not disabled by CTS0* HIGH, only TDRE is inhibited.

DCD0*: Data Carrier DEtect 0 (input)

The DCD0* input allows external control (start/stop) of ASCI channel 0 receive operations. When DCD0* is high, channel 0 RDRF bit is held at 0 regardless of whether the RDR0 (Receive Data Register) is full or empty. the error flags (PE, FE and OVRN bits) are also held at 0. Even after the DCD0* input goes low, these bits will not resume normal operation until the status register (STAT0) is read. This first read of STAT0, while enabling normal operation, will still indicate the DCD0* input is high (DCD0* bit = 1) even though it has gone low, so the STAT0 register should be read twice to ensure the DCD0* bit is reset = 0.

RTS0*: Request to Send (output)

RTS0* allows the ASCI to start and stop another communication device transmission (for example, by connection to that device CTS* input). RTS0* is a 1 bit output port, having no side effects on other ASCI registers or flags.

CTS1*: Clear to Send 1 (input)

Channel 1 CTS1* input is multiplexed with the RXS pins (Clocked Serial Receive Data). The CTS1* function is selected when the CTS1E bit in STAT1 is set = 1. When enabled, the CTS1* operation is equivalent to CTS0*. Modem control signal timing is shown in Fig 27 and Fig 28.

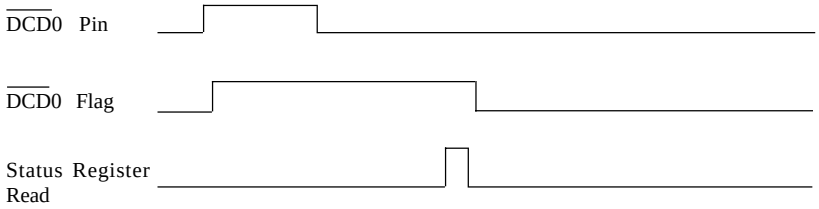


Figure 27 **DCD0 Timing**

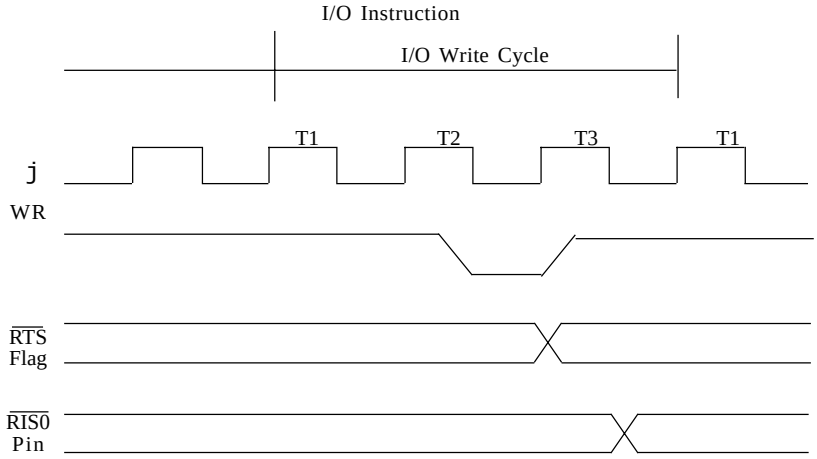


Figure 28 **RTS0 Timing**

ASCI Interrupts

Fig. 29 shows the ASCI interrupt request generation circuit.

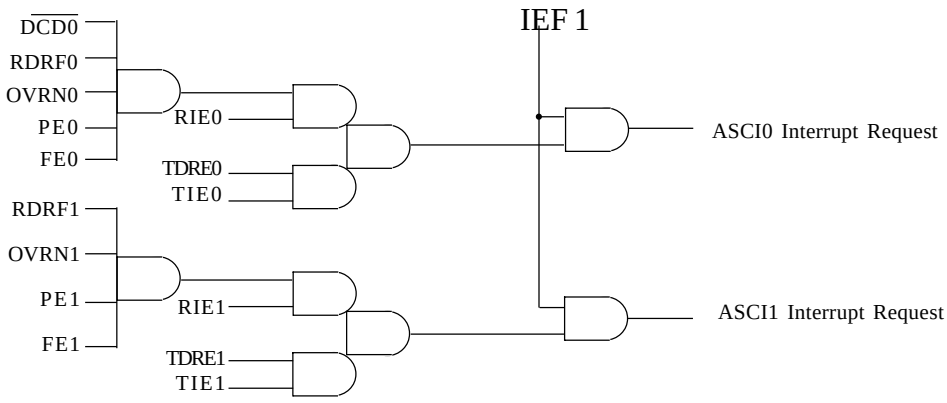


Figure 29 **ASCI Interrupt Request Circuit Diagram**

ASCI « DMAC Operation

Operation of the ASCI with the on-chip DMAC channel 0 requires the DMAC to be correctly configured to utilise the ASCI flags as DMA request signals.

ASCI and RESET

During RESET, the ASCI status and controls registers are initialised as defined in the individual register descriptions.

Receive and Transmit operation are stopped during RESET, but the contents of the transmit and receive data register (TDR and RDR) are not changed by RESET.

Direct Memory Access (DMA)

DMA Controller (DMAC)
The AB181E-20- contains a two channel DMA (Direct Memory Access) controller which supports high speed data transfer. Both channels (channel 0 and channel 1) have the following capabilities.

Memory Address Space
Memory source and destination addresses can be specified anywhere within the 1M bytes physical address space using 20-bit source and destination memory addresses. Memory transfer can also cross 64k bytes physical address boundaries arbitrarily, without CPU intervention.

I/O Address Space
I/O source and destination addresses can be directly specified anywhere within the 64k bytes I/O address space (16-bit source and destination I/O addresses).

Transfer Length
Up to 64k bytes can be transferred based on 1 16-bit count register.

DREQ* Length
Level and edge sense DREQ* input detection are selectable.

TEND* Output
Used to indicate DMA completion to external devices.

Transfer Rate
Each byte transfer can occur every three bus cycles. Wait states can be inserted in DMA cycles for slow memory or I/O devices. At the system clock (Ø) = 20MHz, the DMA transfer rate is as high as 6.7 megabytes/second (no wait states).

Additional feature disc for DMA interrupt request by DMA END. Each channel has additional specific capabilities.

Channel 0
Memory « memory, memory « I/O, memory « memory mapped I/O transfers
Memory address increment, decrement, np-change
DMA to and from both ASCI channels
Higher priority then DMAC channel 1

Channel 1
Memory « I/O transfer
Memory address increment, decrement

DMAC Registers
Each channel of the DMAC (channel 0, 1) has three registers specifically associated with that channel.

Channel 0

SAR0	-	Source Address Register
DAR0	-	Destination Address Register
BCR0	-	Byte Count Register

Channel 1

MAR1	-	Memory Address Register
IAR1	-	I/O Address Register
BCR1	-	Byte Count Register

The two channels share three additional registers in common.

DSTAT	-	DMA Status Register
DMODE	-	DMA Mode Register
DCNTL	-	DMA Control Register

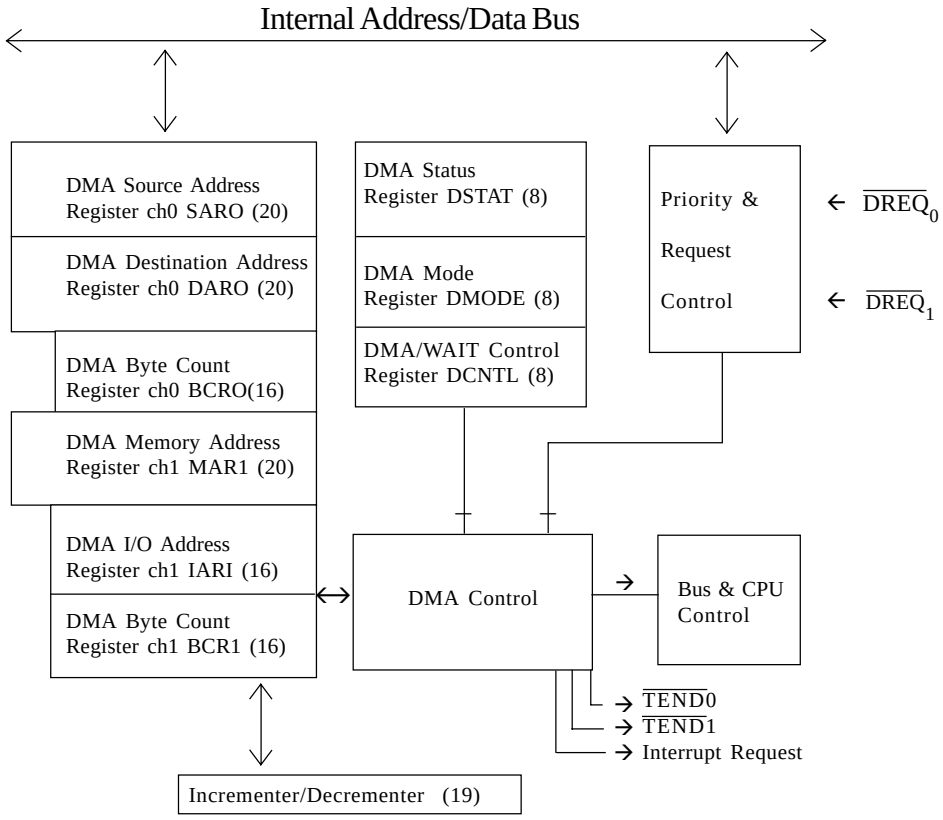


Fig 30 DMAC Block Diagram

DMAC Register Description

Channel 0 Source Address Register (SARO: I/O Address = 20H to 22H)

Specifies the physical source address for channel 0 transfers. The register contains 20 bits and may specify up to 1M bytes memory addresses or up to 64k bytes I/O addresses. Channel 0 source can be memory, I/O or memory mapped I/O.

Channel 0 Destination Address Register (DAR0: I/O Address = 23H to 25H)

Specifies the physical destination address for channel 0 transfers. The register contains 20 bits and may specify up to 1M bytes memory addresses or up to 64k bytes I/O addresses. Channel 0 destination can be memory, I/O or memory mapped I/O.

Channel 0 Bytes Count Register (BCR0: I/O Address = 26H to 27H)

Specifies the number of bytes to be transferred. This register contains 16 bits and may specify up to 64k bytes transfers. When one byte is transferred, the register is decremented by one. If "n" bytes should be transferred, "n" must be stored before the DMA operation.

Channel 1 Memory Address Register (MAR1: I/O Address = 28H to 2AH)

Specifies the physical memory address for channel 1 transfers. This may be destination of source memory address. This register contains 20 bits and may specify up to 1M bytes memory addresses.

Channel 1 I/O Address Register (IAR1: I/O Address = 2BH to 2CH)
Specifies the I/O address for channel 1 transfers. This may be destination or source I/O address. This register contains 16 bits and may specify up to 64k bytes I/O addresses.

Channel 1 Byte Count Register (BCR1: I/O Address = 2EH to 2FH)
Specifies the number of bytes to be transferred. This register contains 16 bits and may specify up to 64k bytes transfers. When one byte is transferred, the register is decremented by one.

DMA Status Register (DSTAT)
DSTAT is used to enable and disable DMA transfer and DMA termination interrupts. DSTAT also allows determining the status of a DMA transfer (in progress or completed).

DMA Status Register (DSTAT : I/O Address = 30H)

bit	7	6	5	4	3	2	1	0
	DE 1	DE 0	DWE_1	DWE_0	DIE1	DIE0	-	DME
	R/W	R/W	W	W	R/W	R/W		R

DE1: DMA Enable Channel 1 (bit 7)
When DE1 = 1 and DME = 1, channel 1 DMA is enabled. When a DMA transfer terminates (BCR1 = 0), DE1 is reset to 0 by the DMAC. When DE1 = 0 and the DMA interrupt is enabled (DIE1 = 1), a DMA interrupt request is made to the CPU.

To perform a software write to DE1, DWE1* should be written with 0 during the same register write access. Writing DE1 = 0 disables channel 1 DMA, but DMA is restartable. Writing DE1 = 1 enables channel 1 DMA and automatically sets DME (DMA Main Enable) = 1. DE1 is cleared = 0 during RESET.

DEO: DMA Enable Channel 0 (bit 6)
When DE0 = 1 and DME = 1, channel 0 DMA is enabled. When a DMA transfer terminates (BCR0 = 0), DEO is reset to 0 by the DMAC. When DEO = 0 and the DMA interrupt is enabled (DIE0 = 1), a DMA interrupt request is made to the CPU.

To perform a software write to DEO, DWE0* should be written with 0 during the same register write access. Writing DEO = 0 disables channel 0 DMA. Writing DEO = 1 enables channel 0 DMA and automatically sets DME (DMA Main Enable) = 1. DE0 is cleared = 0 during RESET.

DWE1*: DE1 Bit Write Enable (bit 5)
When performing any software write to DE1, DWE1* should be written with 0 during the same access. DWE1* write value of 0 is not held and DWE1* is always read as 1.

DWE0*: DEO Bit Write Enable (bit 4)
When performaing any software write to DE0, DWE0* should be written with 0 during the same access. DWE0* write value of 0 is not held and DWE0* is always read as 1.

DIE1: DMA Interrupt Enable Channel 1 (bit 3)
When DIE1 is set = 1, the termination of channel 1 DMA transfer (indicated when DE1 - 0) causes a CPU interrupt request to be generated. When DIE1 = 0, the channel 1 DMA termination interrupt is disabled. DIE1 is cleared = 0 during RESET.

DIE0: DMA Interrupt Enable Channel 0 (bit 2)
When DIE0 is set = 1, the termination channel 0 of DMA transfer (indicated when DE0 = 0) causes a CPU interrupt request to be generated. When DIE0 = 0, the channel 0 DMA termination interrupt is disabled. DIE0 is cleared = 0 during RESET.

DME: DMA Main Enable (bit 0)
A DMA operation is only enabled when its DE bit (DE0 for channel 0, DE1 for channel 1) and the DME bit are set =1.
When NMI occurs, DME is reset = 0, thus disabling DMA activity during the NMI interrupt service routine. To restart DMA, DE0 and/or DE1 should be written with 1 (even if the contents are already 1). This automatically sets DME = 1, allowing DMA operations to continue. Note that DME cannot be directly written. It is cleared = 0 by NMI or indirectly set = 1 by setting DE0 and /or DE1 = 1. DME is cleared = 0 during RESET.

DMA Mode Register (DMODE)
DMODE is used to set the addressing and transfer mode for channel 0.

DMA Mode Register (DMODE : I/O Address = 31H)

bit	7	6	5	4	3	2	1	0
	-	-	DM1	DM0	SM1	SM0	MMOD	-
			R/W	R/W	R/W	R/W		

DM1, DM0: Destination Mode Channel 0 (bits 5-4)
Specifies whether the destination for channel 0 transfers is memory, I/O or memory mapped I/O and the corresponding address modifier. DM1 and DM0 are cleared = 0 during RESET.

DM1	DM0	Memory/ I/O	Address Increment/Decrement
0	0	Memory	+1
0	1	Memory	-1
1	0	Memory	fixed
1	1	I/O	fixed

SM1, SM0: Source Mode Channel 0 (bits 3-2)
Specifies whether the source for channel 0 transfers is memory, I/O or memory mapped I/O and the corresponding address modifier. SM1 and SM0 are cleared = 0 during RESET.

SM1	SM0	Memory/ I/O	Address Increment/Decrement
0	0	Memory	+1
0	1	Memory	-1
1	0	Memory	fixed
1	1	I/O	fixed

The following Table shows all DMA transfer mode combinations of DM0, DM1, SM0, SM1. Since I/O « transfers are not implemented, twelve combinations are available.

DM1	DM0	SM1	SM0	Transfer Mode	Address Increment/Decrement
0	0	0	0	Memory to Memory	SAR+1, DAR+1
0	0	0	1	Memory to Memory	SAR-1, DAR+1
0	0	1	0	Memory mapped I/O to Memory	SAR fixed, DAR+1
0	0	1	1	I/O to Memory	SAR fixed, DAR+1
0	1	0	0	Memory to Memory	SAR+1, DAR-1
0	1	0	1	Memory to Memory	SAR-1, DAR-1
0	1	1	0	Memory mapped I/O to Memory	SAR fixed, DAR-1
0	1	1	1	I/O to Memory	SAR fixed, DAR-1
1	0	0	0	Memory to Memory mapped I/O	SAR+1, DAR fixed
1	0	0	1	Memory to Memory mapped I/O	SAR-1, DAR fixed
1	0	1	0	reserved	
1	0	1	1	reserved	
1	1	0	0	Memory to I/O	SAR+1, DAR fixed
1	1	0	1	Memory to I/O	SAR-1, DAR fixed
1	1	1	0	reserved	
1	1	1	1	reserved	

For channel 0 DMA with I/O source or destination, the DREQ0* input times the transfer.

MMOD DMA Bus Mode

When this bit is written with 1 the DMA takes all available bus cycles in memory-memory mode and program execution is paused until the end of the transfer (Burst Mode). When set to 0, the bus is shared between the execution engine and DMA permitting program execution to continue whilst the transfer takes place (Cycle Steal).

DMA/WAIT Control Register (DCNTL)

DCNTL controls the insertion of wait states into DMAC (and CPU) accesses of I/O and the DMA request mode for each DREQ* (DREQ* and DREQ1*) input is defined as level or edge sense. DCNTL also sets the DMA transfer mode for channel 1, which is limited to memory« I/O transfers.

DMA/WAIT Control Register (DCNTL : I/O Address = 32H)

bit	7	6	5	4	3	2	1	0
	——	——	IWII	IWI0	DMS1	DMS0	DIM1	DIM0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IWI1, IWI0: I/O Wait Insertion (bits 5-4)
Specifies the number of wait states introduced into CPU or DMAC I/O access cycles. IWI1 and IWI0 are set = 11 during RESET (see the *Wait State Control* section under *Memory Management Unit* for details).

DMS1, DMS0: DMA Request Sense (bits 3-2)
DMS1 and DMS0 specify the DMA request sense for channel 0 (DREQ0*) and channel 1 (DREQ1*) respectively. When reset to 0, the input is level sense and when set to 1 the input is edge sense. DMS1 and DMS0 are cleared = 0 RESET.

DIM1, DIM0: DMA Channel 1 I/O and Memory Mode (bits 1-0)
Specifies the source/destination and address modifier for channel 1 memory « I/O transfer modes. IM1 and IM0 are cleared = 0 during RESET.

DM1	DM0	Memory/ I/O	Address Increment/Decrement
0	0	Memory to I/O	MAR+1, IAR fixed
0	1	Memory to I/O	MAR-1, IAR fixed
1	0	I/O to Memory	IAR fixed, MAR+1
1	1	I/O to Memory	IAR fixed, MAR-1

Table 31 Channel 1 Transfer Mode

DMA Operation

This section discusses the three DMA operation modes for channel 0, memory « memory, memory «I/O and memory « memory mapped I/O. It also describes the operation of channel 0 DMA with the on-chip ASCI (Asynchronous Serial Communication Interface), as well as Channel 1 DMA.

Memory « Memory - Channel 0
For memory « memory transfers, the external DREQ0* input is not used for DMA transfer timing. The DMA operation will automatically proceed until termination as shown by byte count (BCR0) = 0.

Perform the following operations to initiate memory memory DMA for channel 0.
(1) Load the memory source and destination addresses into SAR0 and DAR0.
(2) Specify memory memory mode addresses increment/decrement in the SM0, SM1, DM0 and DM1 bits of DMODE.
(3) Load the number of bytes to transfer in BCR0.
(4) Program DE0 = 1 (with DWE0* = 0 in the same access) in DSTAT and the DMA operation will start 1 machine cycle later. If interrupt occurs at the same time, the DIE0 bit should be set = 1.

Memory « I/O (Memory Mapped I/O) - Channel 0
For memory « I/O (and memory « memory mapped I/O) the DREQ0* input is used to time the DMA transfers. In addition, the TEND0* (transfer End) output is used to indicate the last (byte count register, BCR0 = 0) transfer.

The DREQ0* input can be programmed as level or edge sensitive.
When level sense is programmed, the DMA operation begins when DREQ0* is sampled low. If DREQ0* is sampled high, control is relinquished to the AB181E-20-CPU after the next DMA byte transfer.

When edge sense is programmed, DMA operation begins at the falling edge of DREQ0*. If another falling edge is during write cycle, the DMAC continues operating. If an edge is not detected, the CPU is given control after the current byte DMA transfer completes. The CPU will continue operating until a DREQ0* falling is detected at which time the DMA operation will (re) start .

During the transfer for channel 0, the TEND0* output will go low synchronous with the write cycle of the last (BCR0 = 0) DMA transfer .

The DREQ0* and TEND0* pins are programmable multiplexed with the CK AO and CKA1 ASCI clock input/outputs. However, when DMA channel 0 is programmed for memory « I/O (and memory « memory mapped I/O) transfers, the CKA0/DREQ0* pin automatically functions as input pin even if it has been programmed as output pin for CKA0. The CKA01/TEND0* pin functions as output for TEND0* by setting CKA1D = 1 in CNTLA1.

To initiate memory « I/O (and memory « memory mapped I/O) DMA transfer for channel 0, perform the following operations.
(1) Load the memory and I/O or memory I/O source and destination addresses into SAR0 and DAR0. I/O addresses (not memory mapped I/O) are limited to 16 bits (A0-A15). Make sure that bits A16 and A17 are 0 (A18 is a don't care) to correctly enable the external DREQ0* input.
(2) Specify memory « I/O or memory « memory mapped I/O mode and address increment/decrement in the SM0, SM1, DM0 and DM1 bits of DMODE.
(3) Load the number of bytes to transfer in BCR0.
(4) Specify whether DREQ0* is edge or level sense by programming the DMS0 bit of DCNTL.
(5) Enable or disable DMA termination interrupt with the DIE0 bit in DSTAT.
(6) Program DE0 = 1 (with DWE0* = 0 in the same access) in DSTAT and the DMA operation will begin under the control of the DREQ0* input.

Memory « ASCI - Channel 0
Channel 0 has extra capability to support DMA transfer to and from the on-chip two channel ASCI. Here the external DREQ0* input is not used for DMA timing, but the ASCI status bits are used to generate an internal DREQ0*. The TDRE (Transmit Data Register Empty) bit and the RDRF (Receive Data Register Full) bit are used to generate an internal DREQ0* for ASCI transmission and reception respectively.

To initiate memory « ASCI DMA transfer, perform the following operations:

(1) Load the source and destination addresses into SAR0 and DAR0. Specify the I/O (ASCI) address as follows:
Bits A0-A7 should contain the address of the ASCI channel transmitter or receiver (I/O addresses 6H-9H).
Bits A8-A15 should equal 0.
Bits A17-A16 should be set according to the following table to enable use of the appropraite ASCI status bit as an internal DMA request.

SAR18	SAR17	SAR16	DMA Transfer Request
X	0	0	DREQ0
X	0	1	RDRF (ASCI channel 0)
X	1	0	RDRF (ASCI channel 1)
X	1	1	reserved

X: don't care

DAR18	DAR17	DAR16	DMA Transfer Request
X	0	0	DREQ0
X	0	1	TDRE (ASCI channel 0)
X	1	0	TDRE (ASCI channel 1)
X	1	1	reserved

X: don't care

(2) Specify memory I/O transfer mode and address increment/decrement in the SM0, SM1, DM0 and DM1 bits of DMODE.
(3) Load the number of bytes in BCR0.
(4) The DMA request sense mode (DMS0 bit in DCNTL) **must** be specified as 'edge sense'.
(5) Enable or disableDMA termination interrupt with the DIE0 bit in DSTAT.
(6) Program DE0 = 1 (with DWE0* = 0 in the same access) in DSTAT and the DMA operation with the ASCI will begin under control of the ASCI generated internal DMA request.
The ASCI receiver or transmitter being used for DMS must be initialised to allow the first DMA transfer to begin. The ASCI receiver must be empty as shown by RDRF = 0.
The ASCI transmitter must be full as shown by TDRE = 0. The first byte should be written to the ASCI Transmit Data Register under program control. The remaining byte will be transferred using DMA.

Channel 1 DMA
DMA channel 1 can perform memory « I/O transfers. Except for different registers and status/control bits, operation is exactly the same as described for channel 0 memory « I/O DMA.

To initiate DMA channel 1 memory « I/O operation perform the following operations:
(1) Load the memory address (19 bits) into MAR1.
(2) Load the I/O address (16 bits) into IAR1.
(3) Program the source/destination and address increment/decrement mode using the DIM1 and DIM0 bit in DCNTL.
(4) Specify whether DREQ1* is level or edge sense in the DMS1 bit in DCNTL.
(5) Enable or disable DMA termination interrupt with the DIE1 bit in DSTAT.
(6) Program DE1 = 1 (with DWE1* = 0 in the same access) in DSTAT and the DMA operation with the external I/O device will begin using the external DREQ1* input and TEND1* output.

DMA BUS Timing

When memory (and memory mapped I/O) is specified as a source or destination, ME* goes low during the memory access. When I/O is specified as s source or destination, IOE* goes low during the I/O access.

When I/O (and memory mapped I/O) is specified as a source or destination, the DMA timing is controlled by the external DREQ* input and the TEND* output indicates DMA termination. The external I/O devices may not overlap addresses with internal I/O and control register, even using DMA.

Wait states can be inserted by programming the on-chip wait state generator or using the external WAIT* input.

For memory to memory transfer (channel 0 only), the external DREQ0* input is ignored.

DMAC Channel Priority

For simultaneous DREQ* request, channel 0 has priority over channel 1. When channel 0 is performing a memory memory transfer, channel 1 cannot operate until the channel 0 operation has terminated. If channel 1 is operating, channel 0 cannot operate until channel 1 releases control of the bus.

DMAC and BUSREQ*, BUSACK*

The BUSREQ* and BUSACK* inputs allow another bus master to take control of the AB181E-20 bus. BUSREQ* and BUSACK* have priority over the on-chip DMAC and will suspend DMAC operation. The DMAC releases the bus to the external bus master at the breakpoint of the DMAC memory or I/O access. Since a single byte DMAC transfer requires a read write cycle, it is possible for the DMAC to be suspended after the DMAC read, but before the DMAC write. Even in this case, when the external master releases the AB181E-20 bus (BUSREQ* HIGH), the on-chip DMAC will correctly continue the suspended DMA operation.

DMAC Internal Interrupts

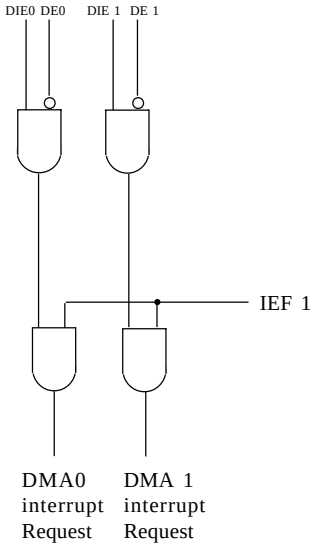


Figure32 DMAC Interrupt Request Circuit Diagram

DEO and DE1 are automatically cleared = 0 by the AB181E-20 at the completion (byte count = 0) of a DMA operation for chanel 0 and channel 1 respectively. They remain 0 until a 1 is written. Since DE0 and DE1 use level sense, an interrupt will occur if the CPU IEF1 is set to 1. So the DMA termination interrupt service routine should disable further DMA interrupt (by programming the channel DIE bit = 0) before enabling CPU interrupts (i.e. IEF1 is set = 1). After reloading the DMAC address and count register, the DIE bit can be set = 1 to reenable the channel interrupt, and at the same time DMA can resume by programming the channel DE bit =1.

DMAC and NMI

NMI, unlike all other interrupts, automatically disables DMAC operation by clearing the DME bit of DSTAT. So the NMI interrupt service routine may respond to time critical events without delay due to DMAC bus usage and NMI can be effectively used as an external DMA abort input, recognising that both channels are suspended by the clearing of DME.

If the falling edge of NMI occurs before the falling clock of the state prior to T3 (T2 or Tw), the DMAC will be suspended and the CPU will start the NMI response at the end of the current cycle.

By setting a channels DE bit = 1, that channels operation can be restarted and DMA will correctly resume from the point at which it was suspended by NMI.

DMAC and RESET

During RESET the bits in DSTAT, DMODE and DCNTL are initialised as stated in their individual register description. Any DMA operation in progress is stopped allowing the CPU to use the bus to perform the REST sequence but the address register (SAR0, DAR0, MAR1, IAR1) and byte count register (BCR0, BCR1) contents are not stopped during RESET.

Programmable Reload Timer (PRT)

The AB181E-20 contains two 16-bit Programmable Reload Timers. Each PRT channel contains a 16-bit down counter and a 16-bit reload register. The down counter can be directly read and written and a down counter overflow interrupt can be programmed as enabled or disabled. PRT channel 1 also has a TOUT output pin (pin 32 - multiplexed with A19) which can be set high or low and toggled, so PRT1 can be programmed to perform output waveform generation.

PRT Block Diagram

Fig 33 shows the PRT block diagram. The two channels have separate timer data and reload registers and a common status/control register. The PRT input clock for both channels is equal to the system clock (ø) divided by 20.

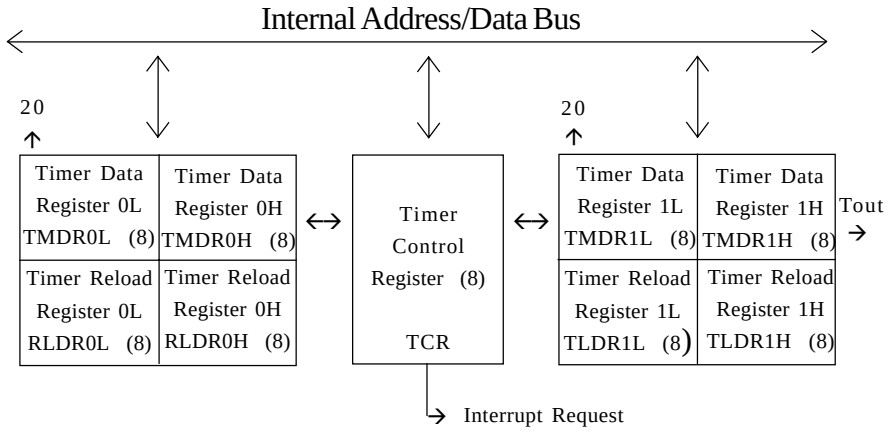


Figure 33 **PRT Block Diagram**

PRT Register Description

Timer Data Register (TMDR: I/O Address = CH0: ODH, OCH CH1: 15H, 14H)

PRT0 and PRT1 each have 16-bit Timber Data Registers (TMDR). TMDR0 and TMDR1 are each accessed as low and high byte registers (TMDR0H, TMDR0L and TMDR1H, TMDR1L). During RESET, TMDR0 and TMDR1 are set = FFFFH.

TMDR is decremented once every twenty ø clocks. When TMDR counts down to 0, it is automatically reloaded with the value contained in the Reload Register (RLDR).

TMDR can be read and written by software using the following procedures. The read procedure uses a PRT internal temporary storage to return accurate data without requiring the timer to be stopped. The write procedure requires that the timer be stopped.

For reading (without stopping the timer), TMDR must be read in the order of lower byte - higher byte (TMDRnL, TMDRnH). The lower byte read (TMDRnL) will store the higher byte value in an internal register. The following higher byte read (TMDRnH) will access this internal register. This procedure ensures timer data validity by eliminating the problem of potential 16-bit timer updating beween each 8 bit read. Note that reading TMDR in higher byte - lower byte order may result in invalid data, and that there are implications for TMDR higher byte internal storage for applications which may read only the lower and/or higher bytes. In normal operation all TMDR read routines should access both the lower and higher bytes, in that order.

For writing, the TMDR down counting must be inhibited using the TDE(Timber Down Count Enable) bits in the TCR (Timer Control Register), following which any or both higher and lower bytes of TMDR can be freely written (and read) inany order.

Reload Register (RLDR: I/O Address = CH0: OEH, OFH CH1: 16H, 17H)

PRT0 and PRT1 each have 16 bit timer Reload Registers (RLDR0). RLDR0 and RLDR1 are each accessed as low and high byte registers (RLDE0H, RLDR0L and RLDR1H, RLDR1L). During RESET RLDR0 and RLDR1 are set = FFFFH.

When the TMDR counts down to 0, it is automatically reloaded with the contents of RLDR.

Timer Control Register

TCR monitors both channel (PRT0, PRT1) TMDR status and controls enabling and disabling of both counting and interrupts as well as controlling the output pin (A19/TOUT-pin 32) for PRT 1.

Timer Control Register (TCR : I/O Address = 10H)

bit	7	6	5	4	3	2	1	0
	TIF1	TIF0	TIE1	TIE0	TOC1	TOC0	TDE1	TDE0
	R	R	R/W	R/W	R/W	R/W	R/W	R/W

TIF1: Timer Interrupt Flag (bit 7)

When TMDR1 decrements to), TIF1 is set = 1. This can generate an interrupt request if enabled by TIE1 = 1. TIF1 is reset = 0 when TCR is read and the higher or lower byte of TMDR1 are read. During RESET, TIF1 is cleared =0.

TIF0: Timer Interrupt Flag (bit 6)

When TMDR0 decrements to 0, TIF0 is set = 1. This can generate an interrupt request of enabled by TIE0 =1. TIF0 is reset = 0 when TCR is read and the higher or lower byte of TMDR0 are read. During RESET, TIF0 is cleared = 0.

TIE1: Timer Interrupt Enable (bit 5)

When TIE1 is set = 1, TIF1 = 1 will generate a CPU interrupt request. When TIE1 is reset = 0, the interrupt request is inhibited. During RESET, TIE1 is cleared = 0.

TIE0: Timer Interrupt Enable (bit 4)

When TIE0 is set = 1, TIF0 = 1 will generate a CPU interrupt request. When TIE0 is reset = 0, the interrupt request is inhibited. During RESET, TIE0 is cleared = 0.

TOC1, 0: Timer Output Control (bits 3-2)

TOC1 and TOC0 control the output of PRT1 using the multiplexed A19/TOUT pin as shown below. During RESET, TOC1 and TOC0 are cleared = 0. This selects the address function for A19/TOUT. By programming TOC1 and TOC0, the A18/TOUT pin can be forced HIGH, LOW or toggled TMDR1 decrements to 0.

TOC1	TOC0	OUTPUT
0	0	Inhibited (A19/TOUT pin is selected as an address output function)
0	1	toggled
1	0	0 (A19/TOUT pin is selected as a Timer output function)
1	1	1

TDE1, 0: Timer Down Count Enable (bits 1-0)

TDE1 and TDE0 enable and disable down counting for TMDR1 and TMDR0 respectively. When TDEn is set = 1, down counting is stopped and TMDRn can be freely read or written TDE1 and TDE0 ar cleared = 0 during RESET and TMDRn will not decrement until TDEn is set = 1.

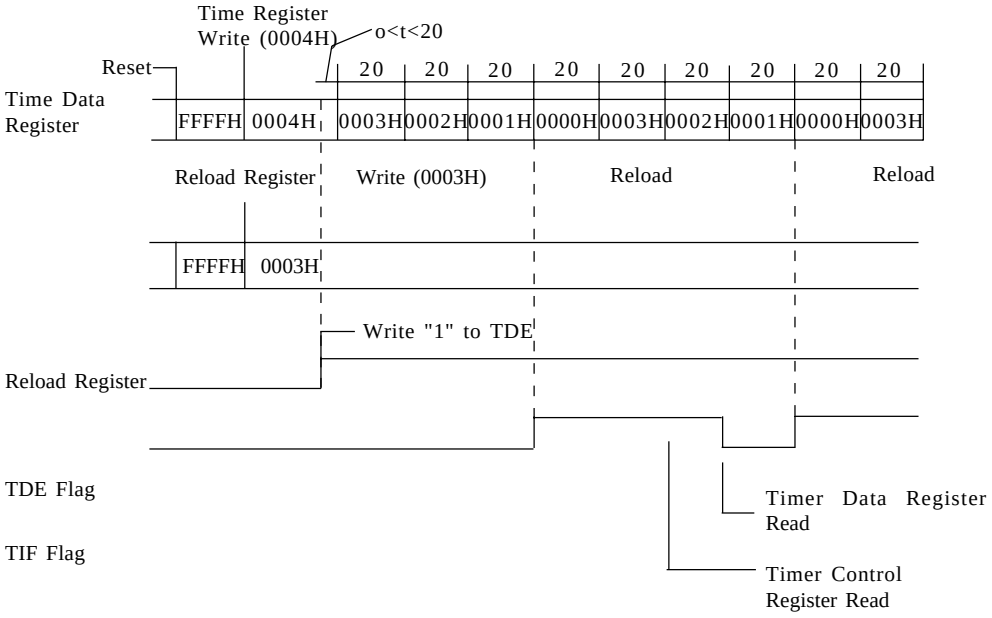


Figure 34 **Timer Operation Timing**

Clocked Serial I/O Port (CSI/O)

The AB181E-20 includes a simple, high-speed clock synchronous serial I/O port. The CSI/O includes transmit/receive (half duplex), fixed 8-bit data and internal or external data clock selection. High-speed operation (baud rate as high as 1M bits/second at fc = 20 MHz) is provided. The CSI/O is ideal for implementing a multiprocessor communication link between the AB181E-20 and single chip controllers as well as additional AB181E-20 CPUs. These secondary devices may typically perform a portion of the system I/O processing such as keyboard scan/decode, LCD interface, etc.

CSI/O Block Diagram

The CSI/O consists of two register - the Transmit/Receive Data Register (TRDR) and Control Register

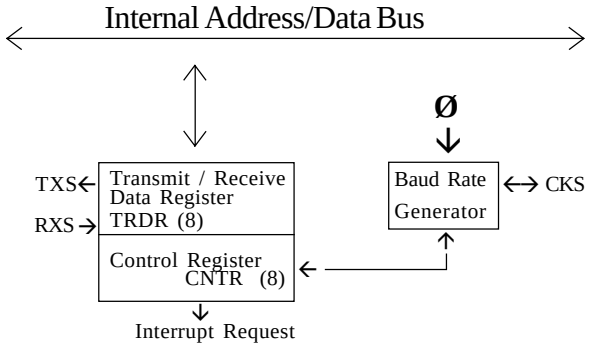


Figure 35 CSI/O Block Diagram

CSI/O Register Description

Transmit/Receive Data Register (TRDR: I/O Address = OBH)

TRDR is used for both CSI/O transmission and reception, so the system design must ensure that the constraints of half-duplex operations are met (Transmit and receive operations can't occur simultaneously). For example, if a CSI/O transmission is attempted at the same time the CSI/O is receiving data, a CSI/O will not work. Because the TRDR is not buffered, attempting to perform a CSI/O transmit while the previous transmit data is still being shifted out causes the shift data to be immediately updated, thereby corrupting the transmit operation in progress. Similarly, reading TRDR while a transmit or receive is in progress should be avoided.

Control/Status Register (CNTR: I/O Address = OAH)

CNTR is used to monitor CSI/O status, enable and disable the CSI/O enable and disable interrupt generation and select the data clock speed and source.

bit	7	6	5	4	3	2	1	0
	EF	EIE	RE	TE	-	SS2	SS1	SS0
	R	R/W	R/W	R/W		R/W	R/W	R/W

EF: End Flag (bit 7)

EF is set = 1 by the CSI/O to indicate completion of an 8 bit data transmit or receive operation. If EIE (End Interrupt Enable) bit =1 when EF is set = 1, a CPU interrupt request will be generated. Program access of TRDR should only occur if EF = 1 to generate a CPU interrupt request. The interrupt request is inhibited if EIE is reset = 0. EIE is cleared. = 0 during RESET.

RE: Receive Enable (bit 5)
A CSI/O receive operation is started by setting RE = 1. When RE is set = 1, the data clock is enabled. In internal clock mode, the data clock is output from the CKS pin. In external clock mode, the clock is input on the CKS pin. In either case, data is shifted in on the RXS pin in synchronisation with the (internal or external) data clock. After receiving 8 bits of data, the CSI/O atuomatically clears RE = 0, EF is set = 1 and an interrupt (if enabled by EIE = 1) will be generated. RE and TE should never both be set =1 at the same time. RE is cleared = 0 during RESET and IOSTOP mode.

Note: the RXS pin (pin 52) is multiplexed with ASCI CTS1* modem control input. In order to enable the RXS function, the CTS1E bit in CNTA1 should be reset = 0.

TE: Transmit Enable (bit 4)
A CSI/O transmit operation is started by setting TE = 1. When TE is set =1, the data block is enabled. In internal clock mode, the data clock is output from the CKS pin. In external clock mode, the clock is input on the CKS pin. In both cases, data is shifted out on the TXS pin synchronous with the (internal or external) data clock. After transmitting 8 bits of data, the CSI/O automatically clears TE =0, EF is set =1 and an interrupt (if enabled by EIE =1) will be generated. TE and RE should never both be set =1 at the same time. TE is cleared = 0 during RESET and IOSTOP mode.

SS2,1, 0: Speed Select 2, 1, 0 (bits 2-0)
SS2, SS1 and SS0 select the CSI/O transmit/receive clock source and speed. SS2, SS1 and SS0 are all set = 1 during RESET.

After RESET, the CKS pin is configured as an external clock input (SS2, SS1, SS0 = 1). Changing these values causes CKS to become an output pin and the selected clock will be output when transmit or receive operations are enabled.

CSI/O Interrupts
The CSI/O interrupt request circuit is shown below .

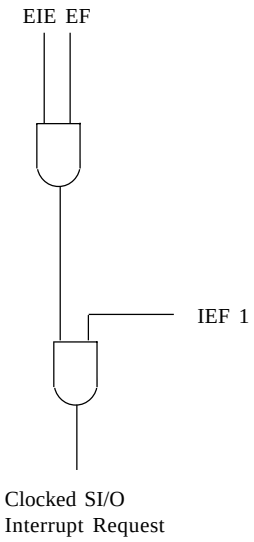


Figure 36 CSI/O Interrupt Circuit Diagram

CSI/O Operation

The CSI/O can be operated using status polling or interrupt driven algorithms.

Transmit - Polling

1. Poll the TE bit in CNTR unitl = 0.
2. Write the transmit data into TRDR.
3. Set the TE bit in CNTR = 1.
4. Repeat 1 to 3 for each transmit data byte.

Transmit - Interrupts

1. Poll the TE bit in CNTR until = 0.
2. Write the first transmit data byte into TRDR.
3. Set the TE and EIE bits in CNTR = 1.
4. When the transmit interrupt occurs, write the next transmit data byte into TRDR.
5. Set the TE bit in CNTR = 1.
6. Repeat 4 to 5 for each transmit data byte.

Receive - Polling

1. Poll the RE bit in CNTR until = 0.
2. Set the RE bit in CNTR = 1.
3. Poll the RE bit in CNTR until = 0.
4. Read the receive data from TRDR.
5. Repeat 2 to 4 for each receive data byte.

Receive - Interrupts

1. Poll the RE bit in CNTR until = 0.
2. Set the RE and EIE bit in CNTR = 1.
3. When the receive interrupts occur read the receive data from TRDR.
4. Set the RE bit in CNTR = 1.
5. Repeat 3 to 4 for each receive data byte.

CSI/O Operation Notes

(1) Disable the transmitter and receiver (TE and RE = 0) before initialising of changing the baud rate. When changing the baud rate after completion of transmission or reception, a delay of a least one bit time is required before baud rate modification.

(2) When RE or TE is cleared = 0 by software, a corresponding receive or transmit operation is immediately terminated. Other than in exceptional circumstances, TE or RE should only be cleared = 0 when EF = 1.

(3) Simultaneous transmission and reception is not possible, so TE and RE should not both be 1 at the same time.

CSI/O and RESET

During RESET each bit in the CNTR is initialised as defined in the CNTR register description. CSI/O transmit and receive operations in progress are aborted during RESET but the contents of TRDR are not changed.

CSI/O Operation Timing Notes

(1) Note that transmitter clocking and receiver sampling timings are different from interntal and external clocking modes. Fig 37 to Fig 40 shows CSI/O Transmit/Receive Timing.

(2) The transmitter and receiver should be disabled (TE and RE = 0) when initialising or changing the baud rate.

Transmit data outputs from the falling edge of the clock up until the next falling edge.

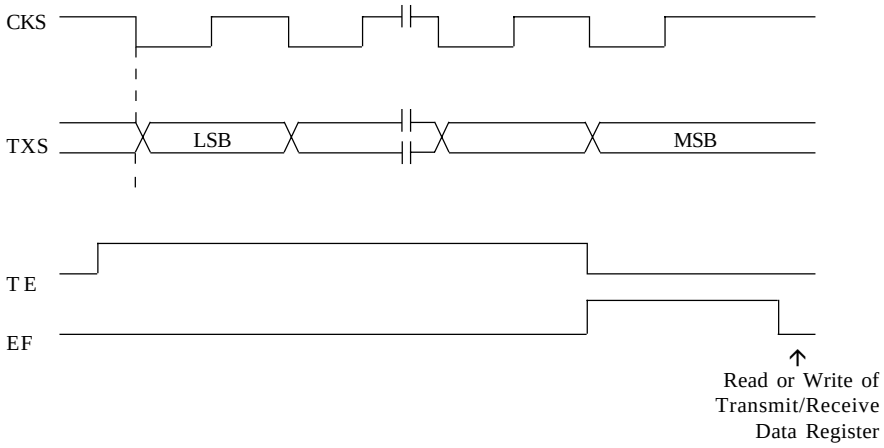


Figure 37 **Transmit Timing - Internal Clock**

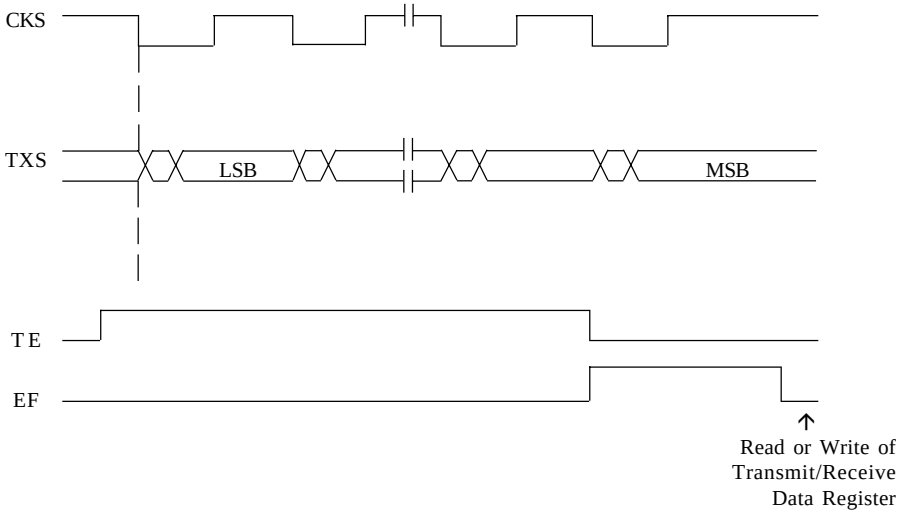


Figure 38 **Transmit Timing - External Clock**

Receive data is latched on to the rising edge of the clock

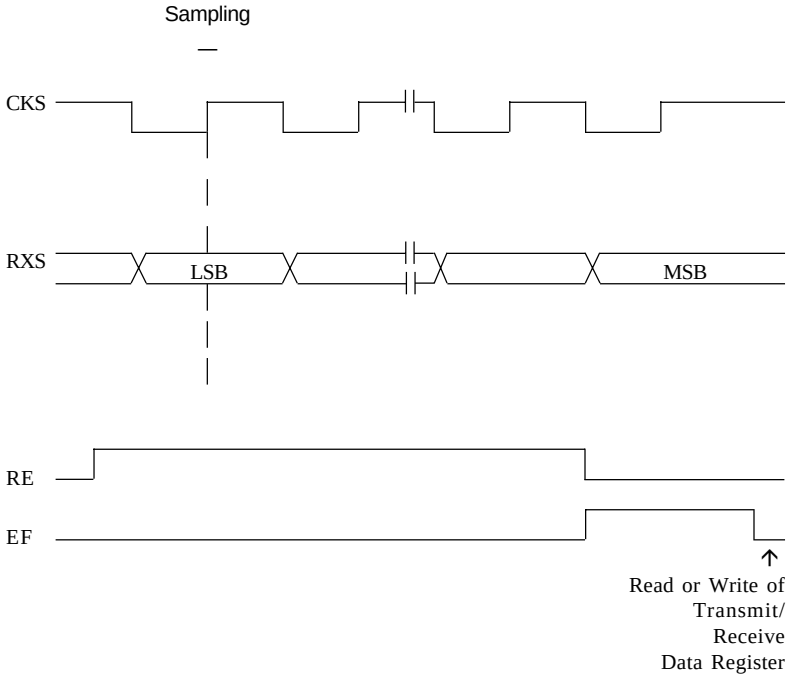


Figure 39 **Receive Timing - Internal Clock**

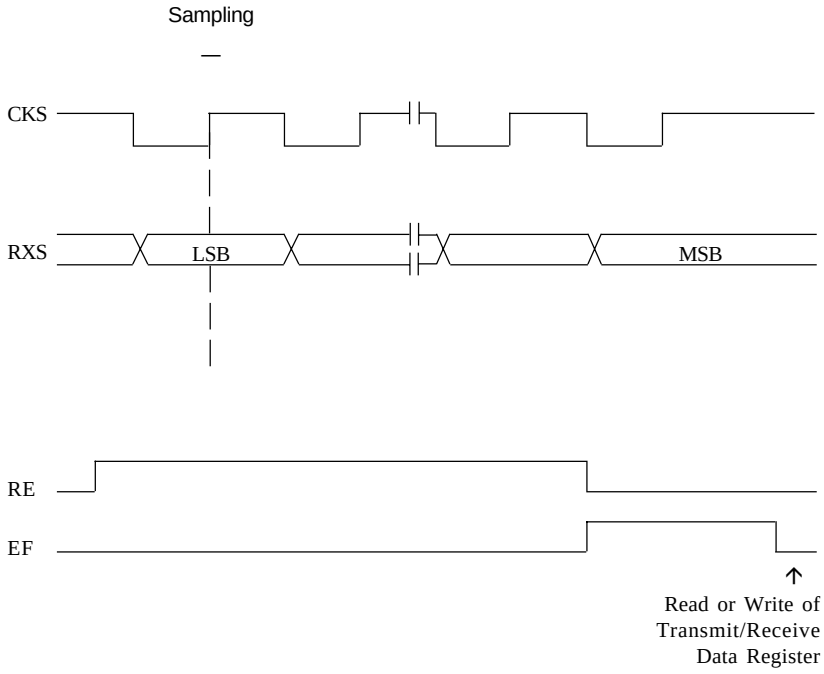


Figure 40 **Receive Timing - External Clock**

Arithmetic Unit

AB181E-20 contains a fixed point arithmetic unit that can perform 32 bit multiplication and accumulation. It is also capable of evaluating the expression:

$$y_{n+1} = y_n + Ax_n + Bx_{n-1} + Cx_{n-2}$$

This type of expression is common in control systems incorporating PID loop filters in the feedback path. All values and results are 32 bits wide, a flag indicating an overflow condition on either the accumulator or multiplier is available. Multiply-accumulate operations take 17 clock cycles and expression evaluation takes 54 clock cycles. A status flag and maskable interrupt are availabe to signal completion of the current operation.

Arithmetic Registers

There are 7 32bit arithmetic registers, these are not directly accessible but can be read and written through an indirect mechanism. These can contain signed numbers in 2's complement form.

Register	Description
Y	Accumulator register In Mult-Acc mode Y <= A * Xn + Y In Expr Eval mode Y <= (A * X_n) + (B * X_{n-1}) + (C * X_{n-2}) + Y This register can be read, preloaded with a value via the Xn register and cleared.
Xn	Current X value, can be loaded and reset by the CPU
Xn+1	Last X value, can be reset but not preloaded.
Xn+2	Last but 1 X value, can be reset but not preloaded.
A	A Coefficient
B	B Coefficient
C	C Coefficient

After each full expression evaluation Xn+1 is copied into Xn+2 and Xn is copied into Xn+1. This does not occur after a multiply-accumulate cycle.

AUCNTRL 3Dh

START	OVERFLOW/ CLEAR-Y	Busy/ CLEAR-X	INTEN	INT	REG SELECT	WORD SELECT

START
Writing a 1 to this bit will start a calculation. This will have no effect if the unit is already performing a calculation but this is inadvisable.

OVERFLOW/CLEAR-Y
When read this bit returns the logical OR of the multiplier and accumulator overflow flags. If either of these units has overflowed a 1 will be returned. Writing a 1 to this bit position will clear the Y_n register.

BUSY/CLEAR-X
When read this bit returns a 1 if the unit is performing a calculation and 0 if it is idle. Writing a 1 to this bit position will clear the X_n , X_{n+1} and X_{n+2} registers.

Note: Setting CLEAR-Y and CLEAR-X simultaneously will perform a LOAD-Y Operation where the contents of X_n are copied into Y.

INTEN
When set to 1 an interrupt will be generated at the end of a calculation, writing 0 Prevents the unit from generating any interrupts.

INT
Set to 1 at the and of a calculation if interrupts are enabled. Write a 0 to this bit to clear this flag.

REG SELECT
These two bits determine which registers are read and written through the AUREGLO and AUREGHI ports.

REGSEL[1]	REGSEL[0]	Register
0	0	A
0	1	B
1	0	C
1	1	x_n

WORD SELECT
When this bit is low, AUREGLO and AUREGHI will allow reads and writes to bytes 0 and 1 of the selected register. When high bytes 2 and 3 are available. This bit is also used to determine the type of operation to be carried out by the unit. Starting a calculation with this bit set low will perform a full expression evaluation if this bit is high a multiply-accumulate only will be performed.

RP_LO 3Bh

-	-	-	-	-	-	-

This port can be used to write to the A,B,C and Xn registers depending upon the setting of the REG SELECT bits in the AUCNTRL register. When read it will return the value in the Y register.

WORD SELECT = 0: Writes to byte 0 of selected arithmetic register. Reads return the value in byte 0 of Y.

WORD SELECT = 1: Writes to byte 2 of selected arithmetic register. Reads return the value in byte 2 of Y.

RP_HI 3Ch

-	-	-	-	-	-	-

This port can be used to write to the A,B,C and Xn registers depending upon the setting of the REG SELECT bits in the AUCNTRL register. When read it will return the value in the Y register.

WORD SELECT = 0: Writes to byte 1 of selected arithmetic register. Reads return the value in byte 1 of Y.

WORD SELECT = 1: Writes to byte 3 of selected arithmetic register. Reads return byte 3 of Y.

Bus Timing Information

Basic CPU Timing

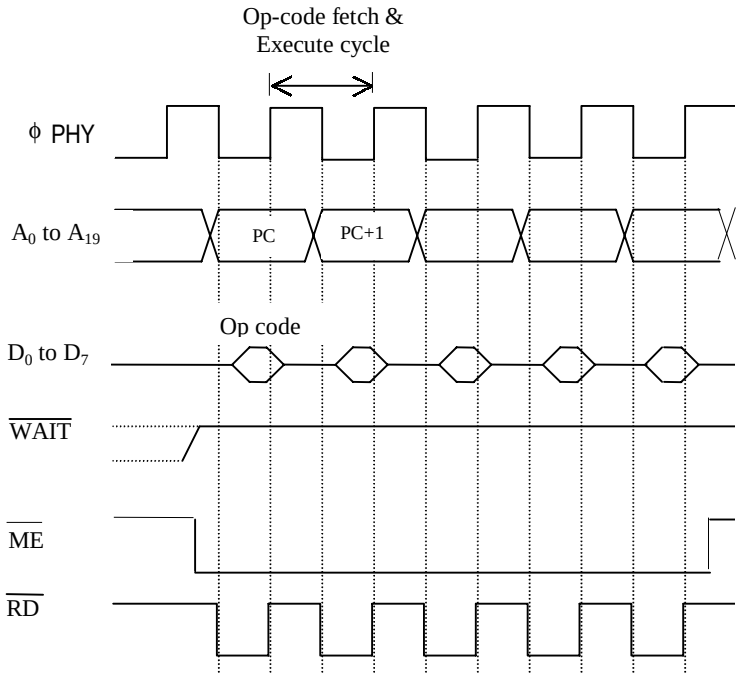


Figure 41

Read Cycle Timing

No wait states

Symbol	Meaning	Min	Nom	Max
Tpp	PHI Clock Period	50	-	-
Tpl	PHI low time	25	-	-
Tph	PHI high time	25	-	-
Tav	PHI high to address valid	16	-	22
Tmea	PHI high to ME asserted	15	-	21
Tmed	PHI high to ME deasserted	16	-	21
Tioa	PHI low to IOE asserted	2	-	21
Tiod	PHI high to IOE deasserted	2	-	21
Trsa	PHI low to RD asserted	2	-	7
Trsd	PHI high to RD deasserted	3	-	8
Tdsu	Data setup time	6	-	-
Tdh	Data hold time	0	-	-

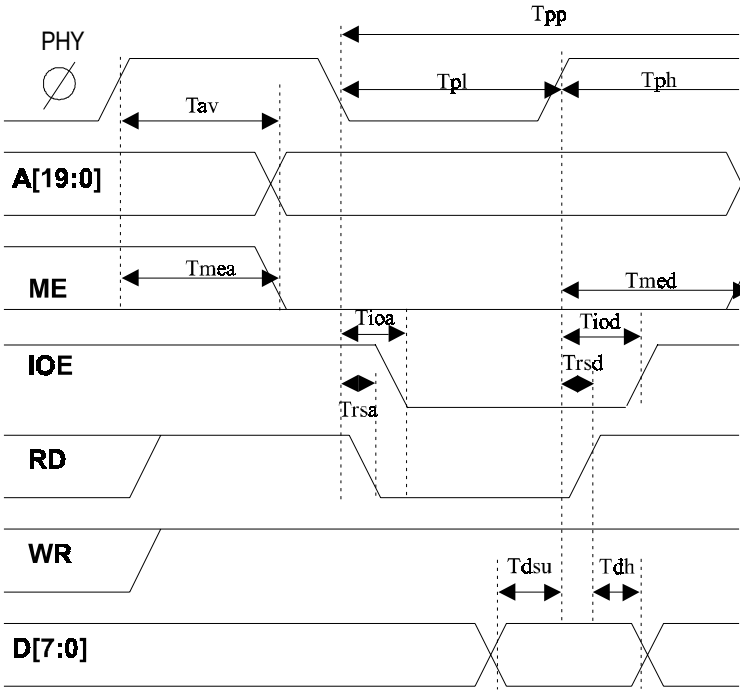


Figure 42

ME is asserted in memory space read cycles, IOE is asserted in I/O space reads.

Write Cycle Timing

No wait states

Symbol	Meaning	Min	Nom	Max
Tpp	PHI Clock Period	50	-	-
Tpl	PHI low time	25	-	-
Tph	PHI high time	25	-	-
Tav	PHI high to address valid	16	-	22
Tmea	PHI high to ME asserted	15	-	21
Tmed	PHI high to ME deasserted	16	-	21
Tioa	PHI high to IOE asserted	15	-	21
Tiod	PHI high to IOE deasserted	15	-	21
Twsa	PHI low to WR asserted	2	-	5
Twsd	PHI high to WR deasserted	2	-	5
Tdv	Data out valid delay time	2	-	6
Tdh	Data out hold time	2	-	-

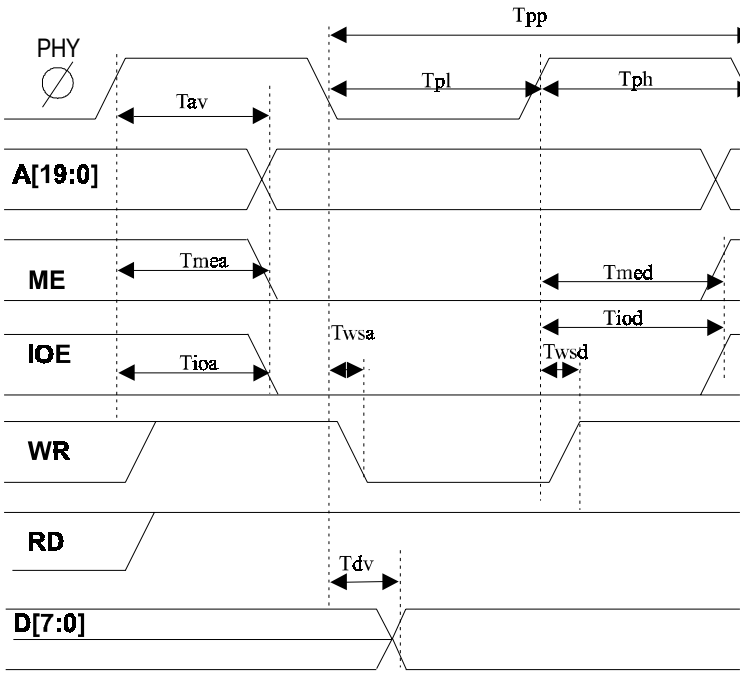


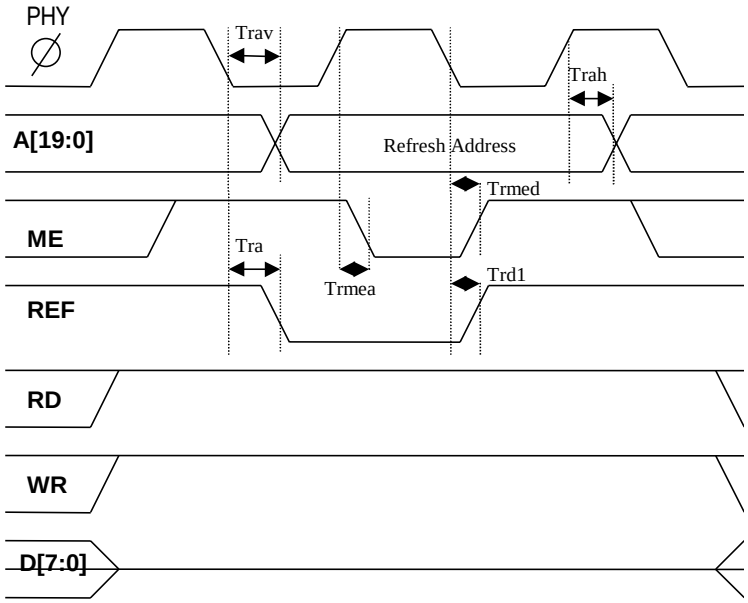
Figure 43

ME is asserted in memory space write cycles, IOE is asserted in I/O space writes.

Refresh Cycle Timing

Two clock

Symbol	Meaning	Min	Nom	Max
Trav	PHI low to refresh address valid	2.5	-	8
Trah	Refresh address hold	2	-	7
Trmea	PHI high to ME asserted	2	-	6
Trmed	PHI low to ME deasserted	2	-	6
Tra	PHI low to REF asserted	2	-	6
Trd1	PHI low to REF deasserted (2 cycle refresh)	3	-	7
Trd2	PHI high to REF deasserted (3 cycle refresh)	2	-	7



Three clock

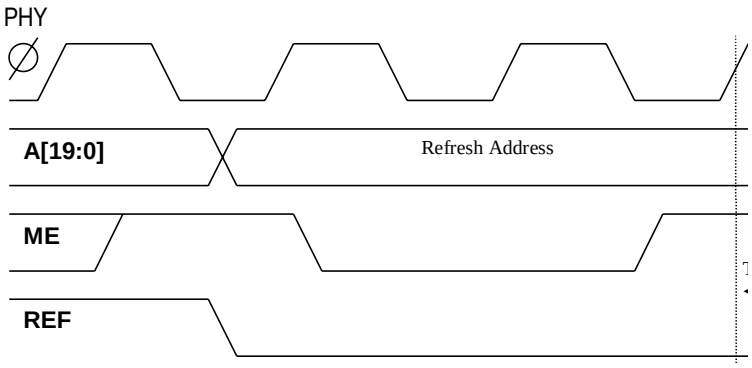


Figure 44

Bus Grant Timing

Symbol	Meaning	Min	Nom	Max
Trba	PHI high to BUSACK asserted	2	-	6
Traz	PHI high to address tristate	2	-	6.5
Trcz	PHI high to control tristate	3	-	6
Trdz	PHI high to data tristate	3	-	10
Tsbd	PHI low to BUSACK deasserted	2	-	8
Tsad	PHI low to address drive	3	-	8
Tscd	PHI low to control drive	3	-	7

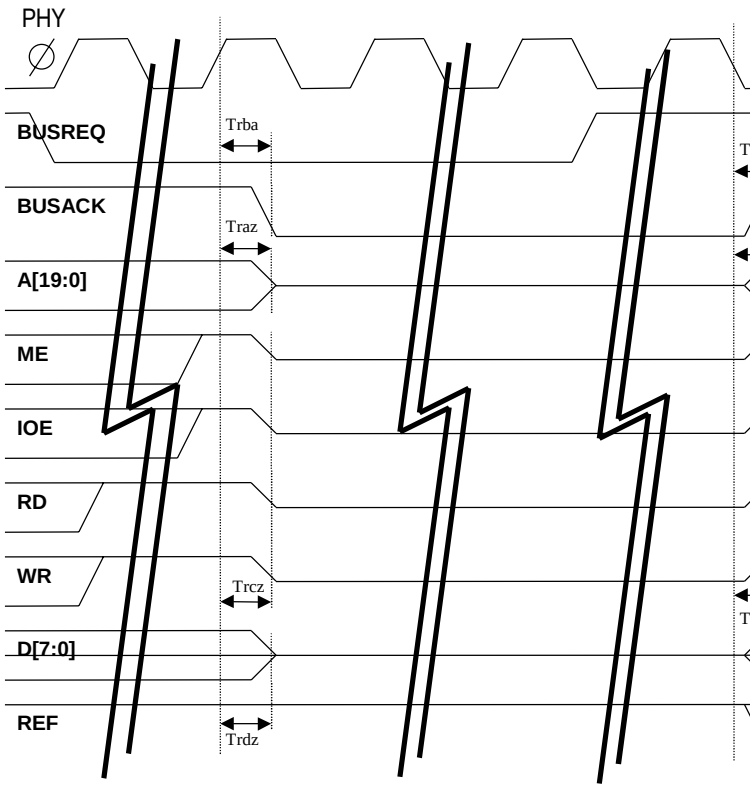
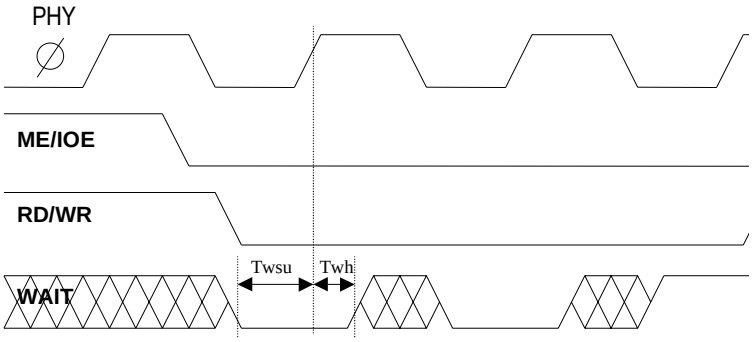


Figure 45

Wait State Generation

Symbol	Meaning	Min	Nom	Max
Twsu	WAIT setup time	20	-	-
Twh	WAIT hold time	3	-	-



Clock generation

The AB181E-20 is provided with an on-chip crystal oscillator and phase locked loop (PLL) which provides frequency multiplication. An external clock can be directly input or the on-chip crystal oscillator can be utilised. The PLL multiplies the oscillator clock frequency by a factor of 8, up to a maximum internal operating frequency of 40MHz. An external clock source or crystal should be used which has a maximum frequency of 5MHz.

Direct Clock Input

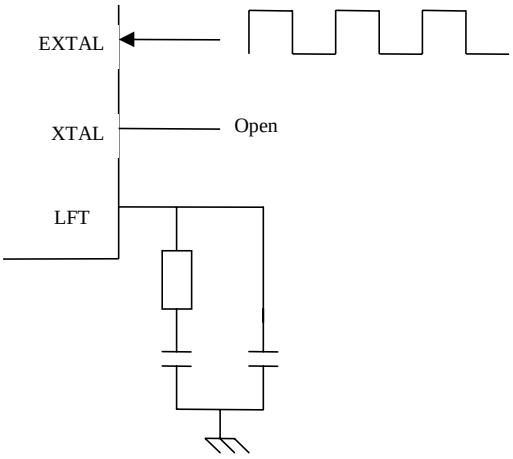


Figure 46

Crystal Oscillator

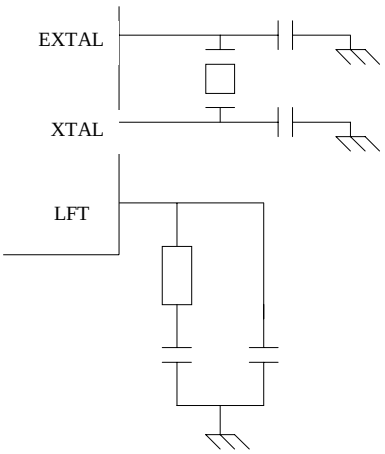


Figure 47

Crystal Oscillator

The AB181E-20 contains an oscillator circuit designed to be used with a 5 MHz AT cut crystal. To use this circuit two load capacitances of 15 pF are required.

When using a crystal it is important to ensure that the noise induced on the traces between the crystal and the package is kept to a minimum. This will require siting the crystal as physically and electrically close to the processor as possible. Trace lengths should be short and no power supplies (VCC) or other signals routed near them. Interference with the clock frequency by electrical noise will cause the PLL watchdog to reset the processor.

PLL Filter Network

The on-chip PLL requires a filter network made up of three components as shown in Figure 48.

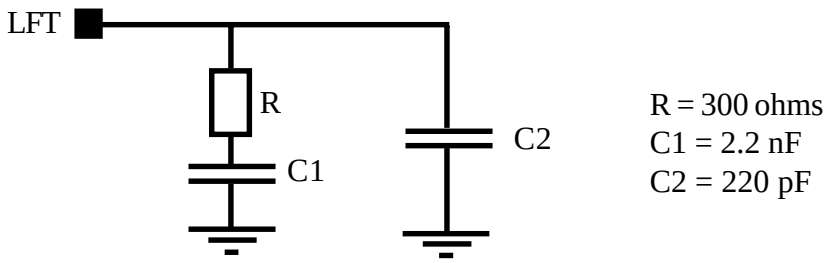


Figure 48

It is important that these components are close to the package pin, that traces are short and not near to other signals. Electrical noise on the LFT pin can cause spurious resets by the PLL watchdog.

External Clock

An externally generated clock of 5 MHz can be applied to the EXTAL pin and with the XTAL pin left open the frequency of PHY will be four times the frequency fed into this pin.

Note: varying the frequency of this waveform will cause the PLL to lose lock and the PLL watchdog will reset the processor.

Appendicies

OP-code Map - Instruction Set

The symbols in the instruction set are explained as follows:

Register
g and g' specify an 8-bit register. ww, xx, yy, and zz specify a pair of 16-bit registers. The symbols and registers correspond as follows:

g, g'	Reg
000	B
001	C
010	D
011	E
100	H
101	L
111	A

ww	Reg
00	BC
01	DE
10	HL
11	SP

xx	Reg
00	BC
01	DE
10	IX
11	SP

yy	Reg
00	BC
01	DE
10	IY
11	SP

zz	Reg
00	BC
01	DE
10	HL
11	AF

Note: suffixed H and L to ww, xx, yy, zz (eg. wwH, IXL) indicate upper and lower 8-bit of the 16-bit register respectively.

Bit
b specifies a bit to be manipulated in the bit manipulation instruction. Bits and b correspond as follows:

b	000	001	010	011	100	101	110	111
Bit	0	1	2	3	4	5	6	7

Condition
f specifies the condition in program control instructions. f and conditions correspond as follows:

f	000	001	010	011	100	101	110	111
Condition	NZ non zero	Z zero	NC non carry	C carry	PO parity odd	PE parity even	P sign plus	M sign minus

Restart Address
v specifies a restart address. v and restart addresses correspond as follows:

v	000	001	010	011	100	101	110	111
Address	00H	08H	10H	18H	20H	28H	30H	38H

Flag
The flag conditions are:

.	not affected
x	undefined
S	set to 1
R	reset to 0
P	Parity
V	overflow

Others

( )M	data in mem address
( )I	data in I/O address
m or n	8-bit data
mn	16-bit data
r	8-bit register
R	16-bit register

b.( )M	content of bit b in mem address
b.gr	" " in register gr
d or j	8-bit signed d'placement
S	source addressing
D	destination add mode
.	AND operation

Appendices: OP-code Map

Table 1 1st Op-code Map Instruction Format :**XX**

				ww(LO=ALL)											LO=0~7				
				BC	DE	HL	SP								BC	DE	HL	AF	zz
				g (LO=0~7)											NZ	NC	PO	P	f
				B	D	H	(HL)	B	D	H	(HL)				00H	10H	20H	30H	v
		HI		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
		LO		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
S (HI=ALL)	B	0000	0	NOP	DJNZ j	JR NZ	JR NC j				NOTE1)					RET f				0
	C	0001	1	LD ww, mn												POP zz				1
	D	0010	2	LD(ww), A		LD (mn) ,HL	LD (mn) , A									JP f, mn				2
	E	0011	3	INC ww				LD g, s				ADD A , s	SUB s	AND s	OR s	JP mn	OUT (m) , A	EX(SP) , HL	DI	3
	H	0100	4	INC g			NOTE1)									CALL f, mn				4
	L	0101	5	DEC g			NOTE1)									PUSH zz				5
	(HL)	0110	6	LD g, m			NOTE1)	NOTE2)			HALT	NOTE2)	NOTE2)	NOTE2)	NOTE2)	ADD A,m	SUB m	AND m	OR m	6
	A	0111	7	RLCA	RLA	DAA	SCF									RST v				7
	B	1000	8	EXAF,AF'	JR j	JR Z, j	JR C, j									RET f				8
	C	1001	9	ADD HL, ww												RET	EXX	JP (HL)	LD SP, HL	9
	D	1010	A	LD A, (WW)		LD HL, (mn)	LD A, (mn)									JP f, mn				A
	E	1011	B	DEC ww				LD g, s				ADC A ,s	SBC A ,s	XOR s	CP s	Table2	IN A, (m)	EXDE,HL	EI	B
	H	1100	C	INC g												CALL f, mn				C
	L	1101	D	DEC g												CALL mn	NOTE3)	Table3	NOTE3)	D
	(HL)	1110	E	LD g, m				NOTE2)				NOTE2)	NOTE2)	NOTE2)	NOTE2)	ADC A,m	SBC A,m	XOR m	CP m	E
	A	1111	F	RRCA	RRA	CPL	CCF									RST v				F
				0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
				C	E	L	A	C	E	L	A					Z	C	PE	M	f
				g(LO=8~F)												08H	18H	28H	38H	v
																LO=8~F				

NOTE 1) (HL) replaces g.

2) (HL) replaces s.

3) If DDH is supplemented as 1st op-code for the instructions which have HL or (HL) as an operand in Table 1, the instructions are executed replacing HL with IX and (HL) with (IX+d).

Example. 22H : LD (mn), HL

DDH 22H : LD (mn), IX

If FDH is supplemented as 1st op-code for the instructions which have HL or (HL) as an operand in Table 1, the instructions are executed replacing HL with IY and (HL) with (IY+d).

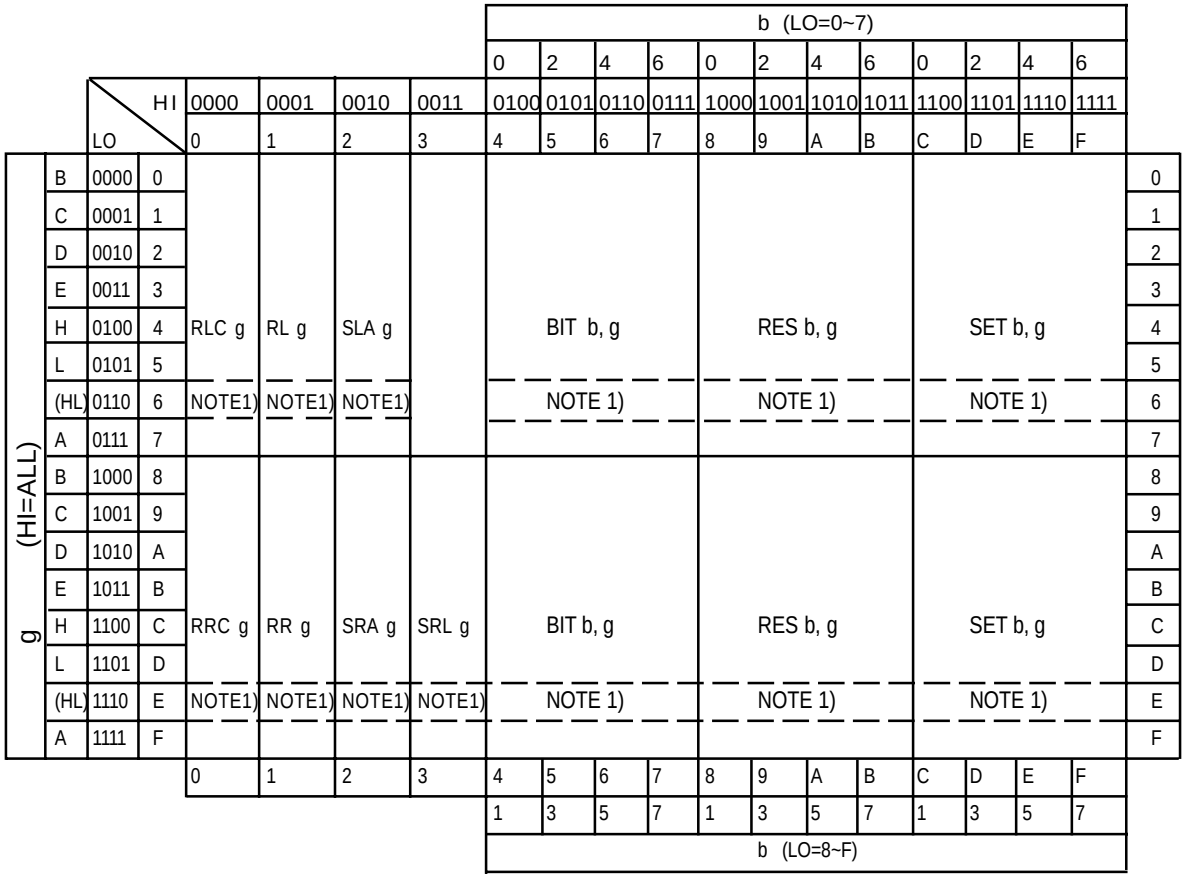
Example 34H : INC (HL)

FDH 34H : INC (IY+d)

However, JP (HL) and EX DE, HL are exception and note the followings.

If DDH is supplemented as 1st op-code for JP (HL), (IX) replaces (HL) as operand and JP (IX) is executed. If FDH is supplemented as1st op-code for JP (HL), (IY) replaces (HL) as operand and JP (IY) is executed. Even if DDH or FDH is supplemented as 1st op-code for EX DE, HL, HL is not replaced and the instruction is regarded as illegal instruction.

Table 2 2nd Op-code map Instruction Format : **CB XX**



Note 1) If DDH is supplemented as 1st op-code for the instructions which have (HL) as operand in Table 2, the instructions are executed replacing (HL) with (IX+d).

If FDH is supplemented as 1st op-code for the instructions which have (HL) as operand in Table 2, the instructions are executed relacing (HL) with (IY+d).

Table 3 2nd Op-code map Instruction Format : ED XX

		ww(LO=ALL)															
		BC	DE	HL	SP												
		g (LO=0~7)															
		B	D	H		B	D	H									
HI	LO	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	INO g, (m)				IN g, (C)						LDI	LDIR				0
0001	1	OUTO (m), g				OUT (C), g						CPI	CPIR				1
0010	2					SBC HL, ww						INI	INIR				2
0011	3					LD (mn), ww				OTIM	OTIMR	OUTI	OUTIR				3
0100	4	TST g			TST (HL)	NEG			TST m	TSTIO m							4
0101	5					RETN											5
0110	6					NOP	NOP										6
0111	7					LD I, A	LD A, I	RRD									7
1000	8	INO g, (m)				IN g, (C)						LDD	LDDR				8
1001	9	OUTO (m), g				OUT (C), g						CPD	CPDR				9
1010	A					ADC HL, ww						IND	INDR				A
1011	B					LD ww, (mn)				OTDM	OTDMR	OUTD	OTDR				B
1100	C	TST g				MLT ww											C
1101	D					RETI											D
1110	E						NOP										E
1111	F					LD R, A	LD A, R	RLD									F
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
		C	E	L	A	C	E	L	A								
		g(LO=8~F)															

AB Semicon Limited
AB Semicon House, 62 Victoria Road,
Burgess Hill, West Sussex RH15 9LR United Kingdom
Tel: +44(0) 1444 870408
Fax: +44(0) 1444 870452
Email: info@hbmuk.com

AB Semicon Inc.
8305 Highway 71 West, Austin, Texas 78735 United States
Tel: +1 512 288 6750
Fax: +1 512 288 7676
Email: info@hbmuk.com

Distributed in Japan
Rikei Corporation
1-26-2, Nishi-Shinjuku, Shinjuku-ku, Tokyo 163-05 Japan
Tel: +81-3-3345-2189
Fax: +81-3-3344-3949

Visit the AB Semicon web-site at: http://www.ab-semicon.com