

ATC-10K50E-LVDS-PIO

INDUSTRY PACK

ALTERA

HARDWARE REFERENCE MANUAL

Revision 2.5.1

April 2005

This Document shall not be duplicated, nor its contents used for any purpose, unless express permission has been granted in advance.

ALPHI TECHNOLOGY CORP.

6202 S. Maple Avenue #120

Tempe, AZ 85283 USA

Tel : (480) 838 - 2428

Fax: (480) 838 - 4477

TABLE OF CONTENTS

1	Introduction:	4
1.1	Functional description:	4
1.2	BASIC BLOCK DIAGRAM:	4
1.3	ELECTRICAL BLOCK DIAGRAM:	6
2	I/O BLOCK DIAGRAM	7
3	MAP ADDRESS :	7
3.1	IDSPACE:	8
3.2	IOSPACE:	9
3.2.1	EPM7160- 10K50E dual addressing registers	9
3.2.2	IDEPLD 10Kxx identification part.....	10
3.2.2.1	EPREV EPM7160 revision control	10
3.2.2.2	ICR Initialization Control Register	11
3.2.2.3	ISR Init Status Register	12
3.2.3	10Kxx ALTERA PROG.....	12
3.2.4	IVRx Interrupt Vector Register	13
3.2.5	S485 and nS485.....	13
3.2.6	Driver enable DE[24..1].....	14
3.2.7	Parallel to Serial Conversion with Termination resistance	15
3.2.7.1	SWPG0 [A1/B1-A16/B16].....	16
3.2.7.2	SWPG1 [A17/B17-A24/B24].....	16
3.2.7.3	WR_SW_CTRL.....	16
3.3	INTSPACE	17
3.4	MEMORY SPACE	17
3.4.1	DPR Timing:.....	17
3.5	SRAM interconnection with 10Kxx	18
3.6	10KxxE Memory Request Lines:	19
3.7	IPbus interconnection with 10Kxx	20
4	10Kxx I/O ASSIGNMENT	21
5	10Kxx IO Control line ASSIGNMENT	24
6	JUMPERS AND CONNECTORS LOCATION	25
6.1	Jumper's description:	26
6.1.1	J1 description.....	26
6.1.2	J2 description.....	26
6.1.3	Stand alone application only	26
7	Connectors Description:	27
7.1	IP bus interface P1	28
7.2	I/O Port P2	29
7.3	Passive Serial Port P4: Bit-blaster / Byte-blaster	30
8	CONFIGURATION DEVICES: SERIAL EEPROM	30

9	PPA Passive parallel asynchronous programming	31
10	EXTERNAL CLOCK FOR 10K50E.....	31
11	Timing between EP7160 AND 10K50E	32
12	Additional timing	33
	Figure 1.1: ATC-10Kxx BLOCK DIAGRAM	6
	Figure 2.1: I/O BLOCK DIAGRAM	7
	Figure 6.1: ATC-10Kxx Jumpers and Headers Location	25
	Table 3-1 IDSEL0 SPACE byte content	8
	Table 3-1 IOSPACE MAP	10
	Table 4-2 SRAM Address and Data to 10Kxx connection	18
	Table 4-3 SRAM Data to 10Kxx connection	19
	Table 7-4 Jumper Description	26
	Table 7-5 J1 Description	26
	Table 7-6 J2 Description	26
	Table 8-7 Connectors Description	27
	Table 8-1 P1 IPBUS connector	28
	Table 8-1 P2 I/O connector	29
	Figure 8-1 P4 connector	30
	Table 8-1 P4 Passive Serial Port P4	30

1 Introduction:

1.1 Functional description:

The ATC-10Kxx-PIO module is populated with a FLEX 10Kxx embedded programmable FPGA. A wide range of 10K packages can be selected. Another PLD from Altera (EPM7160) is used to provide all the timings and interface between the IPBUS, FLEX 10Kxx chip and the 512Kbytes of Dual Access Memory.

Key Features are:

- Up to 24 LVDS driver/receiver with disable option.
- Each line can be selected as an I/O line.
- UP TO 512Kbytes of Dual Ported SRAM.
- 8 or 32 MHz clock.
- 2 interrupts and 2 DMA.
- PPA Configuration: Passive Parallel Asynchronous Configuration Via IP Bus.
- Device Configuration: EPC1, EPC2 or EPC1441.
- Passive Serial configuration: Configuration via Master Blaster communication cable or Byte Blaster MV Parallel Port.

The ATC-10Kxx-PIO uses two chips:

- An **EPM7160** chip is used for all timing related to the IPBUS, DPR, FLEX10K50E.
- An **FLEX10KxxEQC240-3** chip.

Interconnection between the DPR and the 10KxxE is provides.

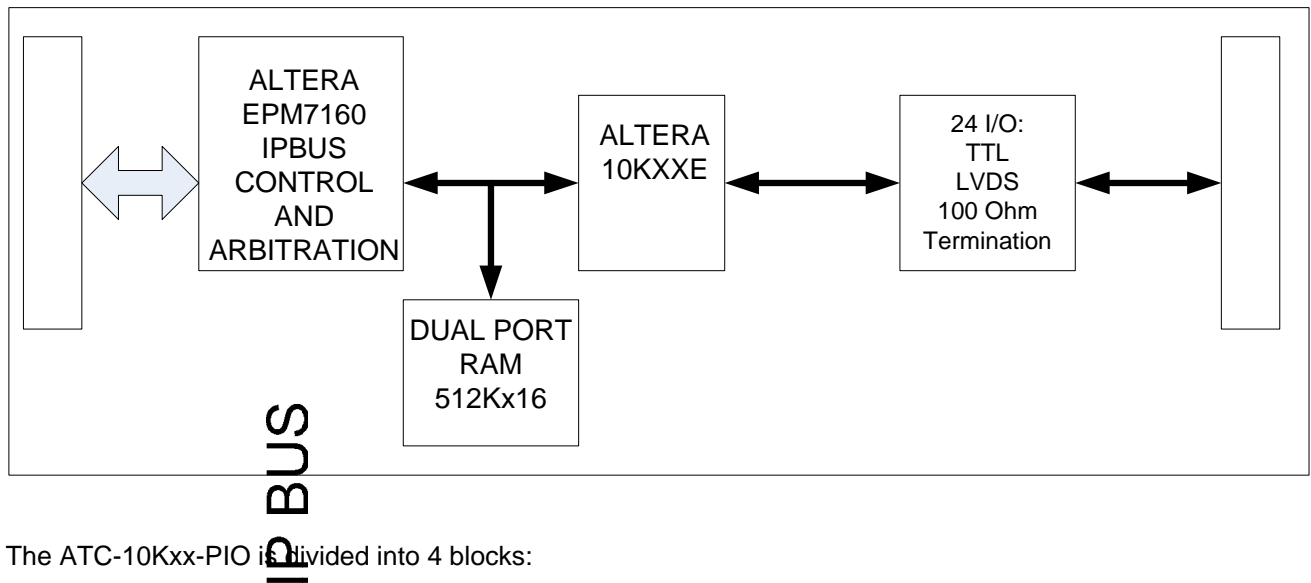
An optional crystal allows the user to run the 10KxxE at a different speed than the IPBUS.

Customer must have ALTERA development tools to implement their own design.

Rev B can be used as stand-alone module.

1. Programming can be achieve through :
 - the IPbus.
 - The byte or Bit blaster.
 - Eprom.

1.2 BASIC BLOCK DIAGRAM:



The ATC-10Kxx-PIO is divided into 4 blocks:

- 1- The EPM7160 E2 based FPGA that is used for the control of the IPBus and arbitration for the Dual Port Ram. The signals (CS,RD,WR) for the DPR are generated by the EPM7160.
- 2- The Dual Port Ram 512kX16 SRAM that can be written and Read from The IpBus and the 10KXXE. The arbitration and the request for the bus has been Taking care of via the FPGA that is programmed by factory
- 3- The Dual Port Ram has been controlled by the FPGA on Board. That process will allow the user to access the ram from the IP bus and the 10KxxE without user intervention with the arbitration. All Arbitration between the IPbus and the 10KxxE has been implemented within the EPM7160.

1.3 ELECTRICAL BLOCK DIAGRAM:

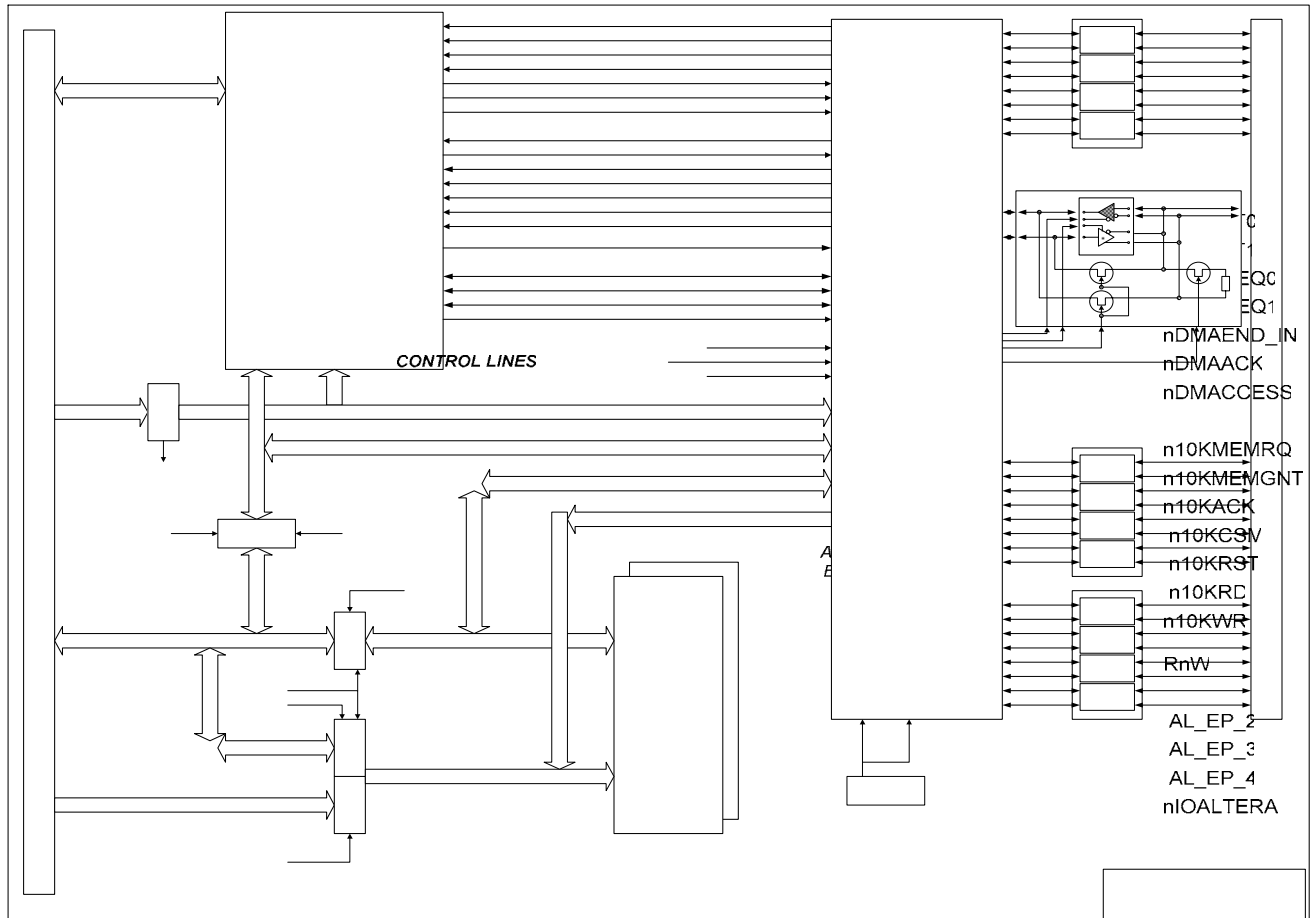
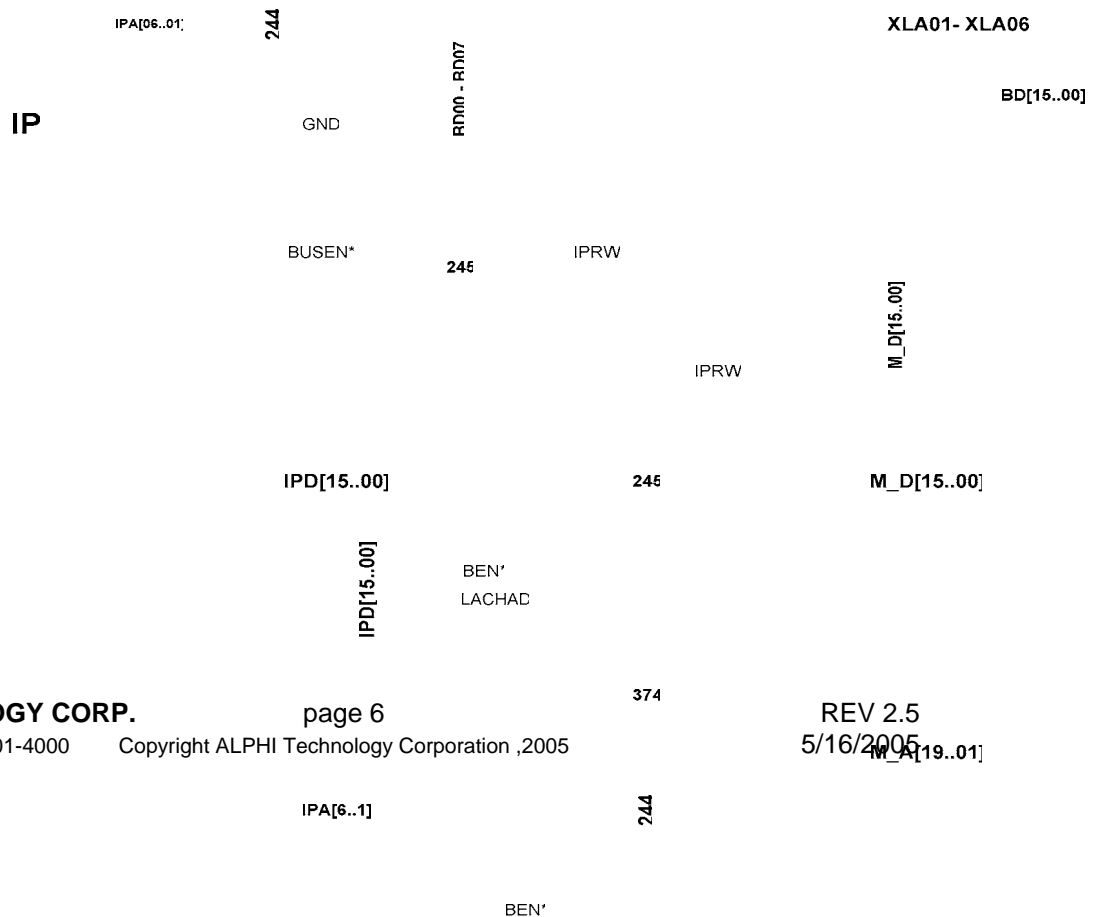
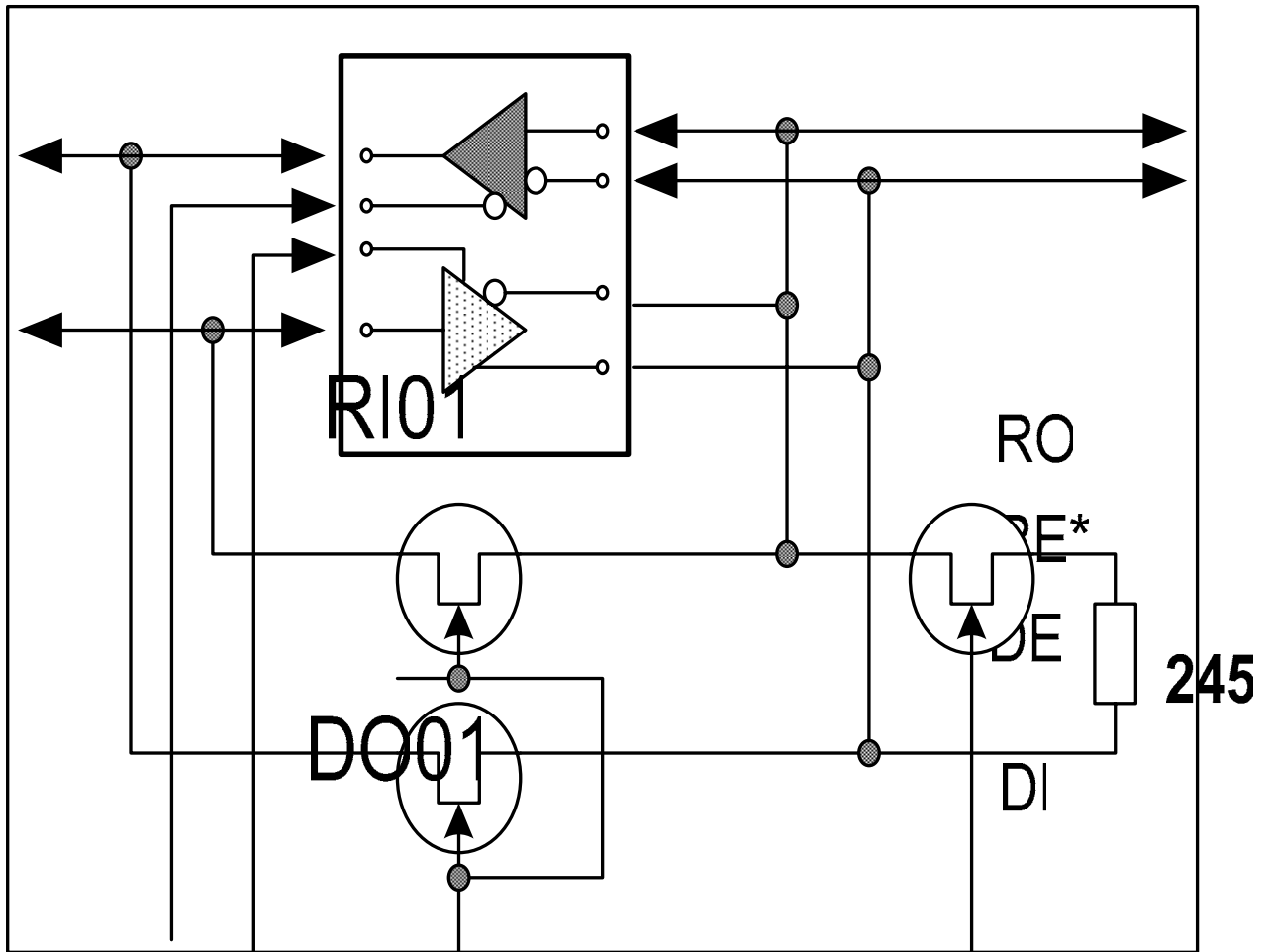


Figure 1.1: ATC-10Kxx BLOCK DIAGRAM



2 I/O BLOCK DIAGRAM



65LVD204A

Figure 2.1: I/O BLOCK DIAGRAM

3 MAP ADDRESS :

The ATC-10Kxx-PIO module uses the three available spaces defined in the Industry Pack specifications.

3.1 IDSPACE:

Up to 32 bytes of registered data provides information about the module to the user. The lower address contains data related to the type of module, revision, etc. Only ODD addresses are valid in byte read mode.

ID space address	Description	Value
\$01	ASCII "I"	\$49
\$03	ASCII "P"	\$50
\$05	ASCII "A"	\$41
\$07	ASCII "H"	\$48
\$09	Manufacturer identification	\$11
\$0B	Module type	\$19
\$0D	Revision module	\$0A
\$0F	Reserved	\$00
\$11	Driver ID,low byte	
\$13	Driver ID,high byte	
\$15	Number of bytes used	\$0C
\$17	CRC	
\$19-\$3F	User space	

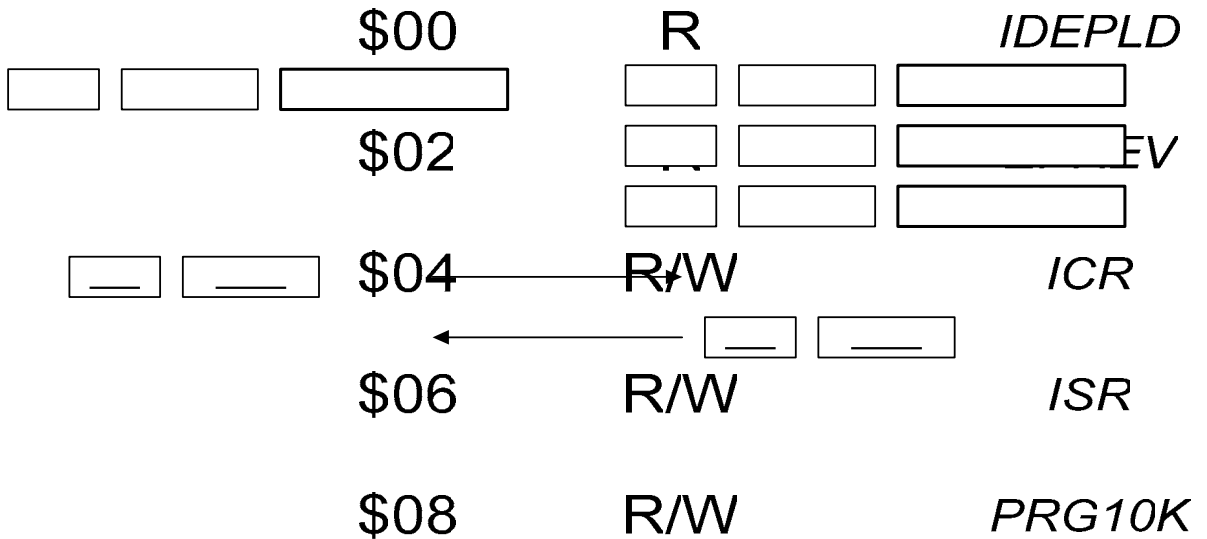
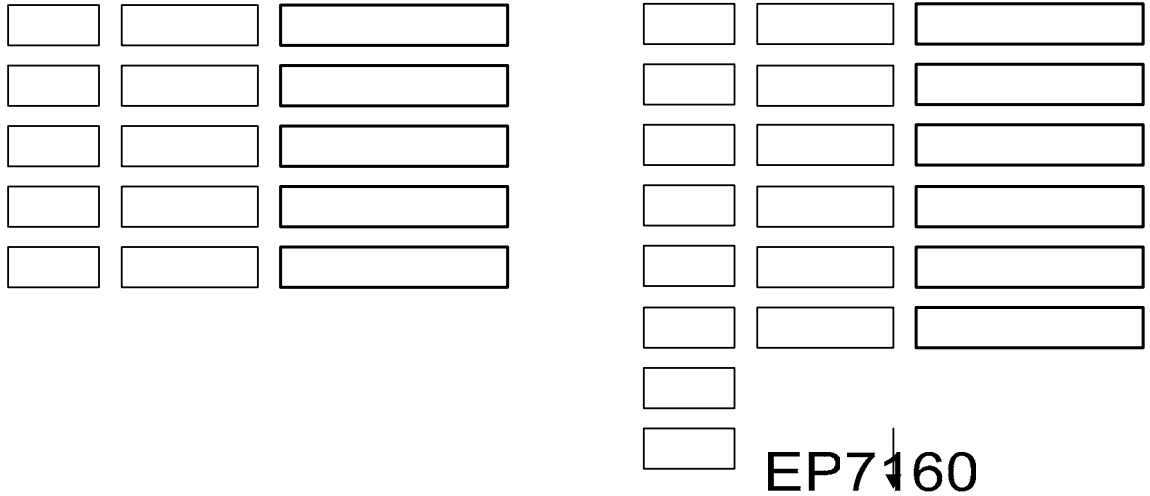
Table 3-1 IDSEL0 SPACE byte content

3.2 IOSPACE:

IP_ALTERA uses the IOSPACE for the following.

Upon IP reset or Power on the register \$08-\$00 located in the EP controller are accessible by the Host for programming purpose. At the end of programming control is given to the 10K50.

3.2.1 EPM7160- 10K50E dual addressing registers



I/O	NAME	REGISTER	TYPE	R/W
\$00	PLD Identification	IDEPLD	Byte	R
\$02	Revision Identification	EPREV	Byte	R
\$04	Init_control_register	ICR	Byte	R/W
\$06	Init status register	ISR	Byte	R
\$08	10Kxx Programming	PRG10K	Byte	W
\$66	Serial Switch Pre-load	SWPG0	Word	R/W
\$68	Serial Switch Pre-load	SWPG1	Byte	R/W
\$6A	Write switch control transfer	WR_SW_CTRL		W
\$78	Interrupt Vector Register #0	IVR0-1	Byte	R/W

Table 3-1 IOSPACE MAP

3.2.2 IDEPLD 10Kxx identification part

Address: IOSPACE + \$00
Actual content: \$02

A read at this location will identify the Altera 10kxxE FPGA implemented on board.

Value	Part #
\$00	10K20
\$01	10K30
\$02	10K50E
\$03	10K100E
\$04	10K130E

3.2.2.1 EPREV EPM7160 revision control

Address: IOSPACE + \$02
Actual content: \$80

A read at this location will identify the revision of the EPM7160. The Bit #7 (when set to a "1") is used to identify if the EPM7160 controller has control of the local bus to program the 10K50. At the end of the programming the signal CONFIG_DONE resets this bit. However it is possible to switch from programming mode to user mode.

A write at the I/O base address \$0 that the user can allocate as a read only ID register into the Altera 10k50 and will set the bit #7 to "1" switching to EPM7160 programming mode.

A \$1 write at the I/O base address + \$2 that correspond to the EPREV register gives back the 10K50 under user control.

Value	Revision
\$00	Rev. A
\$01	Rev. B
\$02	Rev. C

3.2.2.2 ICR Initialization Control Register

Address: IOSPACE + \$04

BD03	BD02	BD01	BD00
-	-	nCONFIG	nCONFIG_EN

BD07	BD06	BD05	BD04
DEV_CLRn	DEV_OE	DEV_CLRn_EN	DEV_OE_EN

Bit 0: nCONFIG_EN:

This bit enables a tri-state buffer that controls the nCONFIG line of the 10kxx upon reset the buffer is tri-state .

Programming of the nCONFIG line can occur only if the MSEL0 and MSEL1 lines are pulled to VCC (remove jumpers from J2).

Bit 1: nCONFIG:

This bit is use to control the nCONFIG line of the 10kxx

Bit 2: Not used

Bit 3:Not used

Bit 4: DEV_OE_EN:

This bit enables a tri-state buffer that control the DEV_OE line of the 10kxx

Bit 5: DEV_CLR_EN:

This bit enables a tri-state buffer that control the DEV_CLRn line of the 10kxx

Bit 6: DEV_OE:

This bit can be use to control the DEV_OE line of the 10kxx

Bit 7: DEV_CLRn:

This bit can be use to control the DEV_CLRn line of the 10kxx

3.2.2.3 ISR Init Status Register

Address: IOSPACE + \$06

This Register provides the status of 10Kxx programming lines

BD03	BD02	BD01	BD00
RDYnBSY	CONF_DONE	nSTATUS	nCONFIG
BD07	BD06	BD05	BD04
Msel1	Msel0	INIT_DONE	CONF_STAT

Bit 1: nCONFIG

This bit when set indicates that the 10Kxx Configuration started. Upon Reset this bit is set pull high by a pull-up resistance to allow multiple possibilities of programming, else this line is low in theory.

Bit 1: nSTATUS

Bit 2: CONF_DONE

Bit 3: RDYnBSY

Bit 4: CONF_STAT

Bit 5: INIT_DONE

Bit 7, 6: MSEL[1..0]

MSEL[1..0]	Description
00	Passive serial Download using Bit blaster
10	Passive parallel synchronous .Not used
11	Passive parallel asynchronous

3.2.3 10Kxx ALTERA PROG

Address: IOSPACE + \$08

The 10Kxx can be programmed in **PPA (parallel passive asynchronous mode)** by writing at the address IOSPACE + \$0.

IPBUS BD00-BD07 are used

For PPA mode Jumper J1 should be set to 2-3. It reconnects DIN (D0) from 10K50E to BD00 (IPBUS).

The 10k50e must used the pin 174 has the BD00 pin for communication with the IPBUS.

When using bit/byte blaster and Epc1 Jumper J1 should be set 1-2.
 DIN is then connected to the jumper pod used by the Bit/Byte blaster.

Details programming of the 10kxx can be found in the application notes AN116 from Altera page 57++.

An example of PPA programming sequence is provided at the end of this manual.

3.2.4 IVRx Interrupt Vector Register

Address: IOWSPACE + \$78

This eight bit register located at address IOWSPACE + \$78 can be read and written by the host. The vector is automatically provided upon INTSPACE cycle perform by the host with lower bit # 0 been the "image" of the interrupt, "0" for interrupt # 0. "1" for interrupt # 1.

3.2.5 S485 and nS485

A set of 6 lines S485[5..0] (12 with the inverted image nS485[5..0]) are use to Enable / Disable the FST3245 switches. The FST3245 is an 8 channels switch with a low Ron (0.3 ohm). Each line control one FST3245, means 8 I/O lines.

At the same time inverted lines (need to be programmed into the Altera) nS485[5..0] enable or disable the differential receiver

Each bit from the S485 register control a set of 4 Ax/Bx groups of lines, means 8 I/O lines. When the S485[x] is set to a "1" the TTL mode is disable. At the same time the receiver of each pair Ax/Bx group is enabling. Four (4) receivers are enabling as differential. For example S485 [0] controls A01/B01, A02/B02, A03/B03, A04/B04 set. The transceiver of each pair is Ax/Bx is disable. Each transceiver can be enable separately.

- A"0" Enable the TTL mode and at the same Time Disable the Differential Driver
- A"1" Disable the TTL mode and at the same Time Enable the Differential Driver

BD07	BD06	BD05	BD04	BD03	BD02	BD01	BD00
-	-	S4855	S4854	S4853	S4852	S4851	S4850

BD15	BD14	BD13	BD12	BD11	BD10	BD09	BD08
-	-	-	-	-	-	-	-

Each Bit controls 8 I/O Line as follow:

Control	I/O	I/O
S4850	A1-A4	B1-B4
S4851	A5-A8	B5-B8
S4852	A9-A12	B9-B12
S4853	A13-A16	B13-B16
S4854	A17-A20	B17-B20
S4855	A21-A24	B21-B24

3.2.6 Driver enable DE[24..1]

Each DEx line is controlled within the 10k50E by the user. An example is to have a:

- WORD DE[16..1]
- BYTE DE[24..17]

The Enable lines control the Output Enable Transmitter of the Differential Driver Max1484. Each Bit controls one Max1484 Transmitter.

A "0" Disable the Differential Driver.

A "1" Enable the Differential Driver.

Pin attribution for the lines is describe farther.

- WORD DE[16..1]

BD07	BD06	BD05	BD04	BD03	BD02	BD01	BD00
DE8	DE7	DE6	DE5	DE4	DE3	DE2	DE1
A8/B8	A7/B7	A6/B6	A5/B5	A4/B4	A3/B3	A2/B2	A1/B1

BD15	BD14	BD13	BD12	BD11	BD10	BD09	BD08
DE16	DE15	DE14	DE13	DE12	DE11	DE10	DE9
A16/B16	A15/B15	A14/B14	A13/B13	A12/B12	A11/B11	A10/B10	A9/B9

- BYTE DE[24..17]

BD07	BD06	BD05	BD04	BD03	BD02	BD01	BD00
DE24	DE23	DE22	DE21	DE20	DE19	DE18	DE17
A24/B24	A23/B23	A22/B22	A21/B21	A20/B20	A19/B19	A18/B18	A17/B17

3.2.7 Parallel to Serial Conversion with Termination resistance

Serial Pre-load register SWPG0-SWPG1

Each pair Ax/Bx has a switch programmable termination resistance of 100 ohms.

The termination resistance should be used with EIA-485 signals located at end point lines. We use maxim MAX335 switches with resistance is between 100 and 200 ohms. Control of the switches is made by a parallel / serial Altera module that transfers 24 bits. Each bit is allocated to one termination switch.

Alphi Technology provides a module that will make a parallel to serial conversion of 24 bits of data pre-loaded into two register.

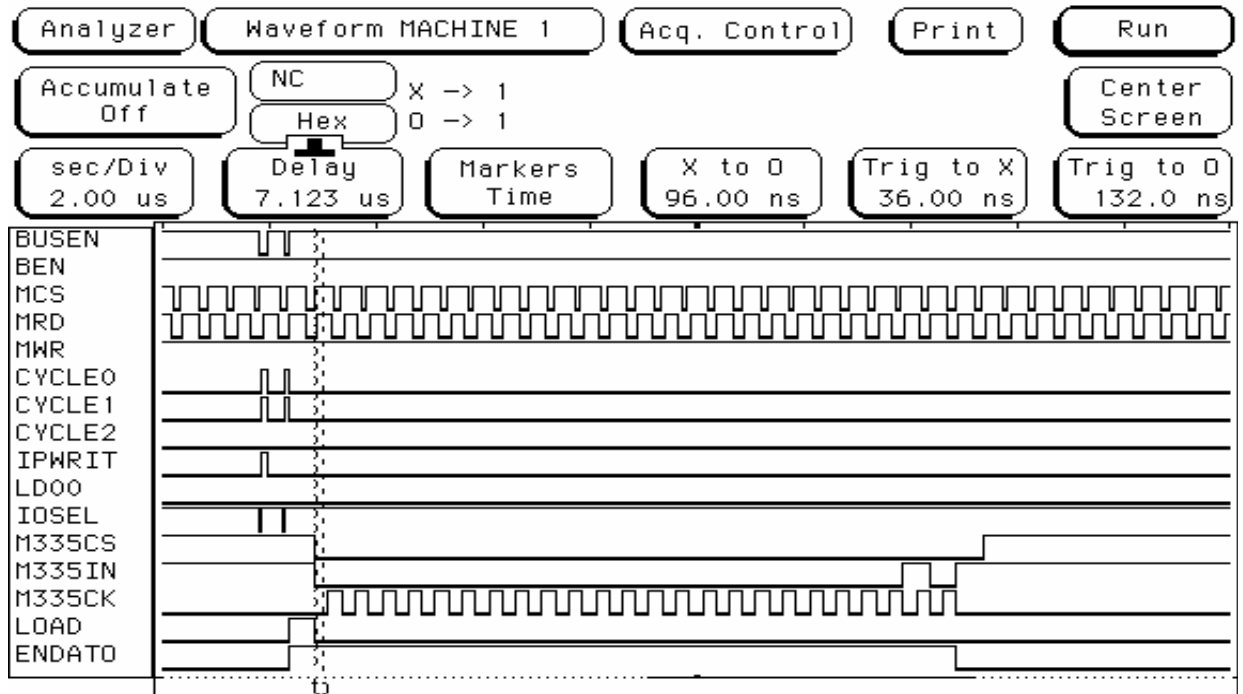
These registers are located at I/O space address + \$66(SWPG0) and \$68(SWPG1).

A "1" will switch the corresponding switch to "on".

A "0" will switch the corresponding switch to "off".

After the data are loaded a write to WR_SW_CTRL, I/O space address + \$6A will serialize the data and transfer them to the MAX335.

Below is an example of bit 1 being transferred on a Analyzer, then an explanation of each register.



3.2.7.1 SWPG0 [A1/B1-A16/B16]

Address: IOSPACE + \$66

- WORD access.

The first register SWPG0 is a 16 bit register for A1/B1 to A16/B16 termination. When turned "ON" Bit set to "1", a 100 Ohm resistor is placed between the corresponding Channel : A1 / B1, A2 / B2.....etc

BD07	BD06	BD05	BD04	BD03	BD02	BD01	BD00
A8/B8	A7/B7	A6/B6	A5/B5	A4/B4	A3/B3	A2/B2	A1/B1

BD15	BD14	BD13	BD12	BD11	BD10	BD09	BD08
A16/B16	A15/B15	A14/B14	A13/B13	A12/B12	A11/B11	A10/B10	A9/B9

3.2.7.2 SWPG1 [A17/B17-A24/B24]

Address: IOSPACE + \$68

- Byte access.

The second register SWPG1 is a 16 bit register but it uses only the lower eight bits for [A17/B17 to A24/B24] termination. When turned "ON" bit set to "1", a 100 Ohm resistor is placed between the corresponding Channel : A17 / B17, A18 / B18.....etc.

BD07	BD06	BD05	BD04	BD03	BD02	BD01	BD00
A24/B24	A23/B23	A22/B22	A21/B21	A20/B20	A19/B19	A18/B18	A17/B17

3.2.7.3 WR_SW_CTRL

Address: IOSPACE + \$6A

- Word or Byte access

Any data written to this location will transfer the 24 stored bits to a 24 bit serial Stream that will turn "On" or "OFF" the corresponding switch to terminate or unterminate the corresponding channel. The programmable switch (Maxim 335) uses a clock of 2 MHz, a minimum delay of 48 us is necessary between changes.

3.3 INTSPACE

When the 10Kxx has an interrupt pending the carrier module can read the IVR register that has been program early by the carrier. The ATC_10Kxx supports one interrupt (IRQ0). Interrupt vector is transferred from the 10Kxx through a buffer to the IPbus. Upon receiving an interrupt cycle (INTSELA) an Interrupt vector register is provided.

3.4 MEMORY SPACE

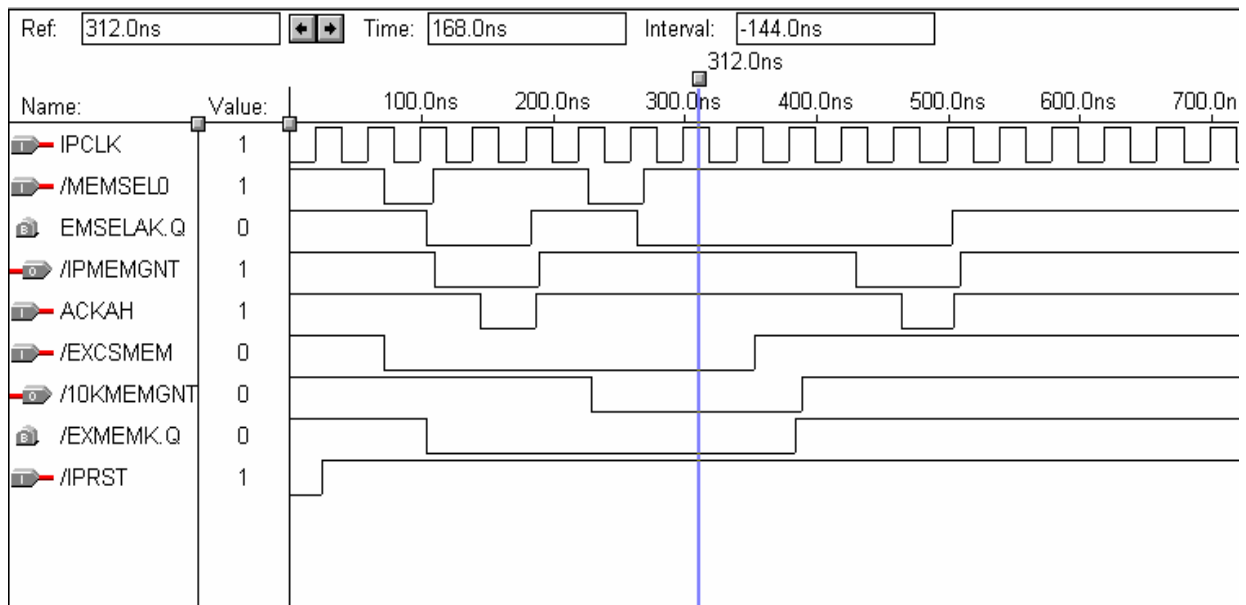
Up to 1Mbytes of SRAM is mapped into the Memory space. The SRAM can be set to be a Dual Access Sram by the IP interface using the Memory space or by the 10Kxx.

The EPM7160 chip provides arbitration between DPR accessed by the I/O connector through the 10Kxx and the IPbus.

An example module to have the 10kxx accessing the Dual Port Ram is provided with timing diagram. Address ,Data lines and controls lines need to be connected to I/O lines for the module to perform an access to the DPR. The EPM7160 provide arbitration and controls lines for the DPR. Some interconnection lines are provided between the two Altera for further application...

3.4.1 DPR Timing:

Figure below show the arbitration timing between an IPBUS access to the DPR and a 10k50e access to the DPR. The IPBUS get the DPR first, then the 10k50e, the following IPBUS DPR access has to wait until the 10k50e has finish the cycle.All the arbitration is made by the EP7160 with IPCLK as reference.



SRAM ADDRESS	10Kxx PIN	Description
M_A01	129	Lower Address Line
M_A02	61	
M_A03	126	
M_A04	127	
M_A05	131	
M_A06	128	
M_A07	132	
M_A08	138	
M_A09	137	
M_A10	134	
M_A11	48	
M_A12	199	
M_A13	105	
M_A14	142	
M_A15	100	
M_A16	148	
M_A17	102	
M_A18	198	
M_A19	103	Higher Address Line

Table 3-2 SRAM Address and Data to 10Kxx connection

3.5 SRAM interconnection with 10Kxx

A separation is provided between the IPbus connected to the 10k50E and the M_d bus connected to the dual ported memory providing faster access. The IPbus can access concurrently registers located within the 10K50E while the 10K50E is accessing the Dual shared memory.

Arbitration between the IPbus accessing the Dual ported memory and the 10k50E accessing the also the Dual ported memory remains the same on the two revision.

SRAM DATA	10Kxx PIN	Description
M_D00	151	Lower Data Line
M_D01	31	
M_D02	74	
M_D03	156	
M_D04	167	
M_D05	106	
M_D06	97	
M_D07	202	
M_D08	99	
M_D09	200	
M_D10	196	
M_D11	158	
M_D12	166	
M_D13	44	
M_D14	78	
M_D15	34	
		Higher Data Line

Table 3-3 SRAM Data to 10Kxx connection

3.6 10KxxE Memory Request Lines:

Signal Name	10Kx x PIN	Description	Type
n10KMEMGNT	109	10k50 receive granted Access to DPR	Input
n10KMEMREQ	194	10K Memory request. Must stay low until end of cycle. Re-sync with IPCLK (GCLK1) before arbitration.	Output
n10KACKB	107	10K Acknowledge. Not used .See Note.	Output
n10KCSM	35	10K Chip select Memory. Feed the EP7160 that will direct the signal to the CS_DPR without delay.	Output
n10KRD	98	10K Read signal for DPR.Same as above	Output
n10KWR	136	10K Write signal for DPR.Same as above	Output
RnW	210	Image of the IPRW signal from the IPbus	Input

Note:This signal is used as acknowledge only with a particular test where the IPBUS read or write to the DPR through the 10k50e using the arbitration process. In this case the EP7160 does not provide the IPack automatically but wait for the n10KACK from the 10k50e. Address I/O base +\$7E is used.

3.7 IPbus interconnection with 10Kxx

Signal Name	10Kxx PIN	Description	Type
nDMAREQ0	195	IP DMA REQ0	Output
nDMAREQ1	108	IP DMA REQ2	Output
nINTREQ0	133	IP INT. REQ 0	Output
nINTREQ1	110	IP INT. REQ 1	Output
nIPDS0	66	DATA STROBE 0	Input
nIPDS1	62	DATA STROBE 1	Input
IPRW	210	IPBUS READ/WRITE	Input
nIPRST	90	IPBUS RESET	Input
IPCLK	91	IPBUS CLOCK 8/32MHZ	Input
XLA01	119	IPBUS ADDRESS LINE	Input
XLA02	184	IPBUS ADDRESS LINE	Input
XLA03	29	IPBUS ADDRESS LINE	Input
XLA04	113	IPBUS ADDRESS LINE	Input
XLA05	116	IPBUS ADDRESS LINE	Input
XLA06	120	IPBUS ADDRESS LINE	Input
nDMAACK	220	IP DMA ACKNOWLEDGE	Input
nDMAEND	218	IP DMA END	Input
nDMAACCES	219	DMA in process	Input

LD00	174
LD01	181
LD02	182
LD03	183
LD04	185
LD05	186
LD06	188
LD07	190
LD08	163
LD09	17
LD10	175
LD11	171
LD12	7
LD13	173
LD14	169
LD15	237

4 10Kxx I/O ASSIGNMENT

CLOCKS:

Signal Name	10Kxx PIN	Description	Type
GCLK1(IPCLK)	91	IPBUS CLOCK (32 OR 8 MHz)	Input
GCLK2(OSC_IN)	211	User Clock if Populated on IC9 location	Input

Note: GCLK2 is also connected to the pin CLKUSER(PIN 11).

See paragraph EXTERNAL CLOCK FOR 10K50E

INTERFACE PIN:

Signal Name	10Kx x PIN	Description	Type
TR_nCS (CS_MAX335)	206	Chip Select Max335 (Switch)	Output
TR_DTA (MAX335_IN)	72	Serial Data for Max335 (Switch)	Output
TR_CLK (MAX335_CLK)	39	Serial Clock for termination Switch (2MHZ)	Output

DE[24..01]

Signal Name	10Kxx PIN	Type
DE[01]	49	Output
DE[02]	215	Output
DE[03]	234	Output
DE[04]	88	Output
DE[05]	71	Output
DE[06]	65	Output
DE[07]	222	Output
DE[08]	87	Output
DE[09]	164	Output
DE[10]	81	Output
DE[11]	228	Output
DE[12]	70	Output
DE[13]	154	Output
DE[14]	82	Output
DE[15]	63	Output
DE[16]	235	Output
DE[17]	64	Output
DE[18]	230	Output
DE[19]	226	Output
DE[20]	67	Output
DE[21]	217	Output
DE[22]	14	Output
DE[23]	79	Output
DE[24]	51	Output

I/O PIN

Altera pin	Name	I/O name	IP connector
153	DO_01	A01	Pin 1
115	RI_01	B01	Pin 2
25	DO_02	A02	Pin 3
149	RI_02	B02	Pin 4
19	DO_03	A03	Pin 5
80	RI_03	B03	Pin 6
13	DO_04	A04	Pin 7
147	RI_04	B04	Pin 8
68	DO_05	A05	Pin 9
203	RI_05	B05	Pin 10
43	DO_06	A06	Pin 11
221	RI_06	B06	Pin 12
117	DO_07	A07	Pin 13
227	RI_07	B07	Pin 14
28	DO_08	A08	Pin 15
201	RI_08	B08	Pin 16
95	DO_09	A09	Pin 17
141	RI_09	B09	Pin 18
152	DO_10	A10	Pin 19
143	RI_10	B10	Pin 20
83	DO_11	A11	Pin 21
18	RI_11	B11	Pin 22
15	DO_12	A12	Pin 23
144	RI_12	B12	Pin 24
46	DO_13	A13	Pin 25
168	RI_13	B13	Pin 26
41	DO_14	A14	Pin 27
86	RI_14	B14	Pin 28
192	DO_15	A15	Pin 29
229	RI_15	B15	Pin 30
24	DO_16	A16	Pin 31
118	RI_16	B16	Pin 32
94	DO_17	A17	Pin 33
204	RI_17	B17	Pin 34
114	DO_18	A18	Pin 35
111	RI_18	B18	Pin 36
21	DO_19	A19	Pin 37
75	RI_19	B19	Pin 38
162	DO_20	A20	Pin 39
146	RI_20	B20	Pin 40
73	DO_21	A21	Pin 41

161	RI_21	B21	Pin 42
9	DO_22	A22	Pin 43
8	RI_22	B22	Pin 44
172	DO_23	A23	Pin 45
12	RI_23	B23	Pin 46
157	DO_24	A24	Pin 47
6	RI_24	B24	Pin 48

5 10Kxx IO Control line ASSIGNMENT

Note: When TTL is selected this signal is generated by the 10KxxE to disable the driver, this signal is the inverse of TTL control signal.

Signal Name	10Kxx PIN	Description	Type
S485_0	45	Enable/Disable TTL Signal for [IO1-08]	Output
S485_1	53	Enable/Disable TTL Signal for [IO9-16]	Output
S485_2	50	Enable/Disable TTL Signal for [IO17-24]	Output
S485_3	56	Enable/Disable TTL Signal for [IO25-32]	Output
S485_4	231	Enable/Disable TTL Signal for [IO33-40]	Output
S485_5	54	Enable/Disable TTL Signal for [IO41-48]	Output

Signal Name	10Kxx PIN	Description	Type
nS485_0	208	Enable/Disable Differential Receiver for [IO01-08]	Output
nIS485_1	223	Enable/Disable Differential Receiver for [IO09-16]	Output
nS485_2	233	Enable/Disable Differential Receiver for [IO17-24]	Output
nS485_3	193	Enable/Disable Differential Receiver for [IO25-32]	Output
nS485_4	55	Enable/Disable Differential Receiver for [IO33-40]	Output
nS485_5	101	Enable/Disable Differential Receiver for [IO41-48]	Output

Spare pins Between FPGA and 10KxxE for custom use:

Signal Name	10Kxx PIN	Description	Type
AL_EP_2	36	TBD	BIDIR
AL_EP_3	38	TBD	BIDIR
AL_EP_4	30	TBD	BIDIR
IOALTERA	191		Input
n10KRST	33	TBD	Output (high)

6 JUMPERS AND CONNECTORS LOCATION

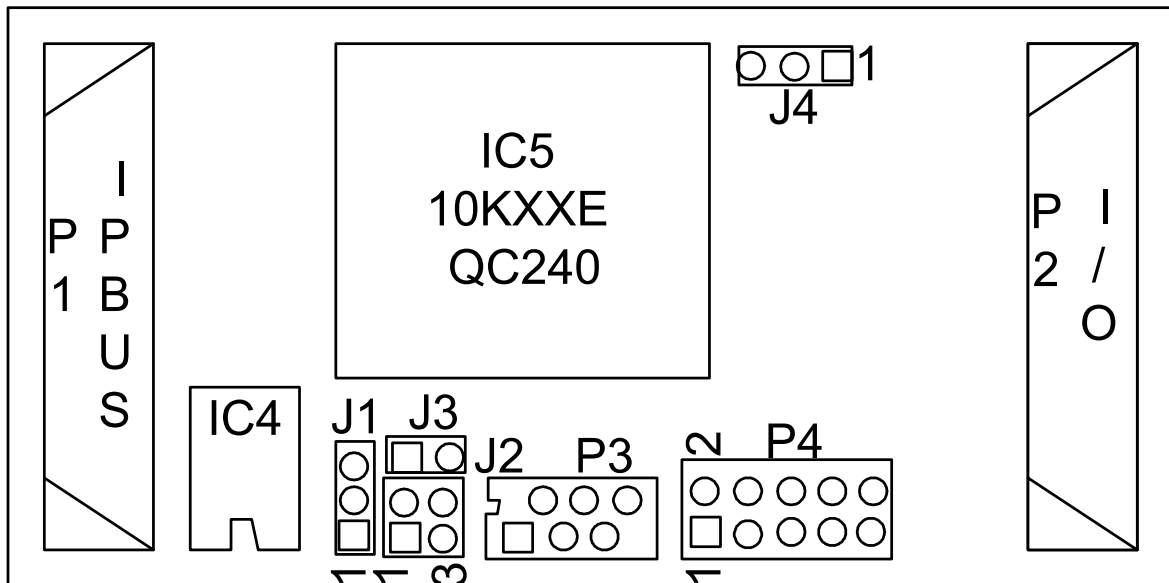


Figure 6.1: ATC-10Kxx Jumpers and Headers Location

6.1 Jumper's description:

Jumpers	Description
J1	10Kxx Din (Data In Selection)
J2	10Kxx Programming Mode selection : MSEL0 and MSEL1
J3	For stand alone use
J4	For factory use
SJ1	Clock source, (Solder side jumper)

Table 6-4 Jumper Description

6.1.1 J1 description

Jumpers	Pin	Description
J1	1-2	When installed 10Kxx Configuration Data source is IC14 (EPC) or Bit/Byte Blaster
J1	2-3	When installed 10Kxx Configuration Data source IP Bus for PPA configuration
J1	1-2	Factory Default (SERIAL DATA IN)

Table 6-5 J1 Description

6.1.2 J2 description

Jumpers	Pin	Description
J2	1-2 ON 3-4 ON	MEL0 & MSEL1 are Pulled Low (Grounded) 10Kxx Configuration Data source Serial
J2	1-2 OFF 3-4 OFF	MEL0 & MSEL1 are Pulled High (VCC) 10Kxx Configuration Data source IP Bus for PPA configuration
J2	1-2 OFF 3-4 ON	MEL0 Low & MSEL1 (VCC) 10Kxx Configuration Data source IP Bus for PPS configuration
J2	1-2 ON 3-4 ON	Factory Default 10Kxx Configuration Data source Serial

Table 6-6 J2 Description

6.1.3 Stand alone application only

J3	1-2 ON	IP stand alone mode
J3	off	Factory Default
SJ1	1-2	IP bus clock source (Factory Default) (this is a solder bridge)
SJ1	2-3	Onboard Osc. source

Table 6-7 J3 Description

7 Connectors Description:

CONNECTOR	DESCRIPTION
P1	IP Bus
P2	I/O Bus
P3	Factory Use : Control PLD Configuration
P4	10Kxx configuration for Bit/Byte Blaster

Table 7-7 Connectors Description

7.1 IP bus interface P1

		P1	
Pin 1	GND	Pin 26	GND
Pin 2	+5V	Pin 27	+5V
Pin 3	IPRESET*	Pin 28	IPRW*
Pin 4	XLD00	Pin 29	IDSEL0*
Pin 5	XLD01	Pin 30	DMAREQ0*
Pin 6	XLD02	Pin 31	MEMSEL0*
Pin 7	XLD03	Pin 32	DMAREQ1*
Pin 8	XLD04	Pin 33	INTESEL0*
Pin 9	XLD05	Pin 34	DMACK*
Pin 10	XLD06	Pin 35	IOSEL0*
Pin 11	XLD07	Pin 36	
Pin 12	XLD08	Pin 37	XLA1
Pin 13	XLD09	Pin 38	DMAEND*
Pin 14	XLD10	Pin 39	XLA02
Pin 15	XLD11	Pin 40	ERROR*
Pin 16	XLD12	Pin 41	XLA03
Pin 17	XLD13	Pin 42	INTREQ0*
Pin 18	XLD14	Pin 43	XLA04
Pin 19	XLD15	Pin 44	INTREQ1*
Pin 20	IPBS0*	Pin 45	XLA05
Pin 21	IPBS1*	Pin 46	Strobe*
Pin 22		Pin 47	XLA06
Pin 23		Pin 48	IPACK*
Pin 24	+5V	Pin 49	+5V
Pin 25	GND	Pin 50	GND

Table 7-1 P1 IPBUS connector

7.2 I/O Port P2

PIN	Name	PIN	Name
Pin 1	A01	Pin 26	B13
Pin 2	B01	Pin 27	A14
Pin 3	A02	Pin 28	B14
Pin 4	B02	Pin 29	A15
Pin 5	A03	Pin 30	B15
Pin 6	B03	Pin 31	A16
Pin 7	A04	Pin 32	B16
Pin 8	B04	Pin 33	A17
Pin 9	A05	Pin 34	B17
Pin 10	B05	Pin 35	A18
Pin 11	A06	Pin 36	B18
Pin 12	B06	Pin 37	A19
Pin 13	A07	Pin 38	B19
Pin 14	B07	Pin 39	A20
Pin 15	A08	Pin 40	B20
Pin 16	B08	Pin 41	A21
Pin 17	A09	Pin 42	B21
Pin 18	B09	Pin 43	A22
Pin 19	A10	Pin 44	B22
Pin 20	B10	Pin 45	A23
Pin 21	A11	Pin 46	B23
Pin 22	B11	Pin 47	A24
Pin 23	A12	Pin 48	B24
Pin 24	B12	Pin 49	
Pin 25	A13	Pin 50	GND

Table 7-1 P2 I/O connector

7.3 Passive Serial Port P4: Bit-blaster / Byte-blaster

The 10Kxx can be programmed using the Bit blaster or the Byte blaster. Connector P4 provides the interface with the ALTERA pod chip. Jumper J1 and J2 should be configured.

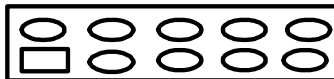


Figure 7-1 P4 connector

Pin	Signal name	Description
1	DCLK	Clock Signal
2	GND	GROUND
3	CONF_DO_N E	Configuration Done
4	VCC	Power supply
5	nCONFIG	Configuration control
6	NC	No connect
7	nSTATUS	Configuration status
8	NC	No connect
9	DATA0	Data to Device
10	GND	Ground

2
1

Table 7-1 P4 Passive Serial Port P4

8 CONFIGURATION DEVICES: SERIAL EEPROM

The 10Kxx can be programmed using Configuration device serial EEPROM: EPC1, EPC2, and EPC1441.

Order Part Number:

EPC1PC8 : Size 1047K Bits
 EPC2PC8 : Size 1696K Bits
 EPC1441PC8: Size 440K Bits

9 PPA Passive parallel asynchronous programming

Download sequence

1. First remove jumpers to have MSEL0 and MSEL1 pull-up to VCC. (J2)
2. Move jumper to have BD00 line connected to D0 of the Altera 10Kxx. (J1)
3. Start with either IP bus reset or power-up reset.
4. The module should be under EP control.
 - Read the revision register at I/O base address+\$02. Bit #7 should be set to "1". With actual revision you should read \$80.
 - Make a write \$0 at I/O base address + \$00. The module switches to EP control allowing access to the programming registers.
5. The ICR initialization register located at I/O base address + \$4 should read "00".
6. Enable the tri-state buffer used to control the nCONFIG line.
 - Write \$01 to the ICR register.
 - This line is normally low if only PPA configuration is allowed, but with this module E2prom or Byte blaster can be used, then the line is Pull-up with a 1k to the VCC. Setting the tri buffer enable pull down the nCONFIG line to low. Should be low for an 8 uS min.
7. Now set the nCONFIG line to a "1" to start the programming process.
8. Read the ISR, Initialization status register and verify that:
 - bit #0 = 1 nCONFIG set to "1"
 - bit #1 = 1 nSTATUS set to "1"
9. Poll the ISR until the RDYnBSY bit #3 is high.
10. Write the first byte to the Initialization Register located at base I/O address + \$8.
11. Check that RDYnBSY line that went low immediately after the write for a maximum of 2 us and is back to high before sending the next data, or wait a minimum of 2 us and send the next data.
12. Repeat steps 9 to 11 until all bits have been downloaded.
 - Note: Rev A: the EPLD controller switch immediately after CONF_DONE is high giving to the 10k50E control preventing to verify that nSTATUS and CONF_DONE status, (It can be done but you have to make a write at address \$02 to give back control to the EPLD, read the register then write at address \$0 to go back to user (10k50E) control.
13. Read the ISR and verify that nSTATUS (bit #1 and CONF_DONE (bit #2) are high
14. Write a \$1 at the I/O base address + \$02 to switch the 10K50E under user control.
15. The EPLD should have release the control of the 10K50, you should be able now to read the ID and revision registers of the 10K50 and also accessed all the other registers.

10 EXTERNAL CLOCK FOR 10K50E

10K50 ALTERA Setup

1. External oscillator.
 - Must assign pin 211 to GCLK2 in block of Altera to have 10K50 use external clock.
2. Internal IP-CLK.
 - Must assign pin 91 to GCLK2 in block of Altera to have 10K50 use internal clock.

ALTERA Compilation

1. Important setup note when compiling using CLKUSR pin of the Altera.
 - Goto **Assign**.

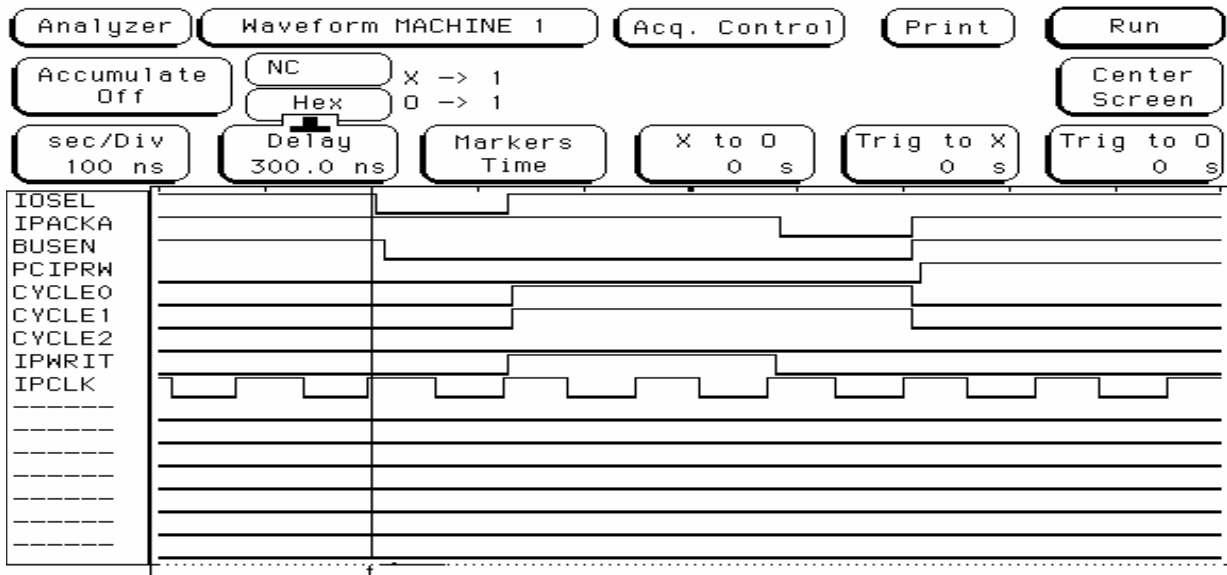
- Then **Device**.
- **Device Options**.
- In the Individual Device Options select the following.
- User-Supplied Start-up Clock (CLKUSR).
- Then bottom of box check on both CLKUSR boxes.

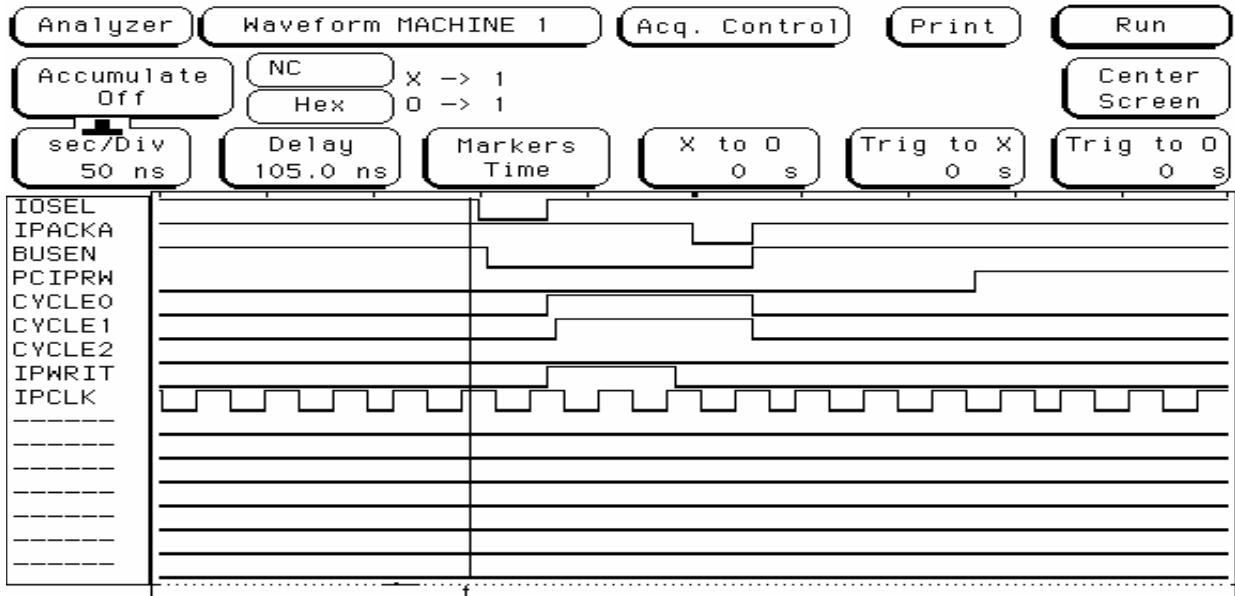
11 Timing between EP7160 AND 10K50E

When the IPBUS access the I/O space register inside the 10k50, the EP7160 provide the following signals.

- CYCLE[2..0] Define the type of cycle
- IP_Write Set to "1" when a write access. This signal is disable upon the EP7160 send the IPACK signal to the IPBUS. The two following timing show the signals at 8MHZ and 32 MHz.

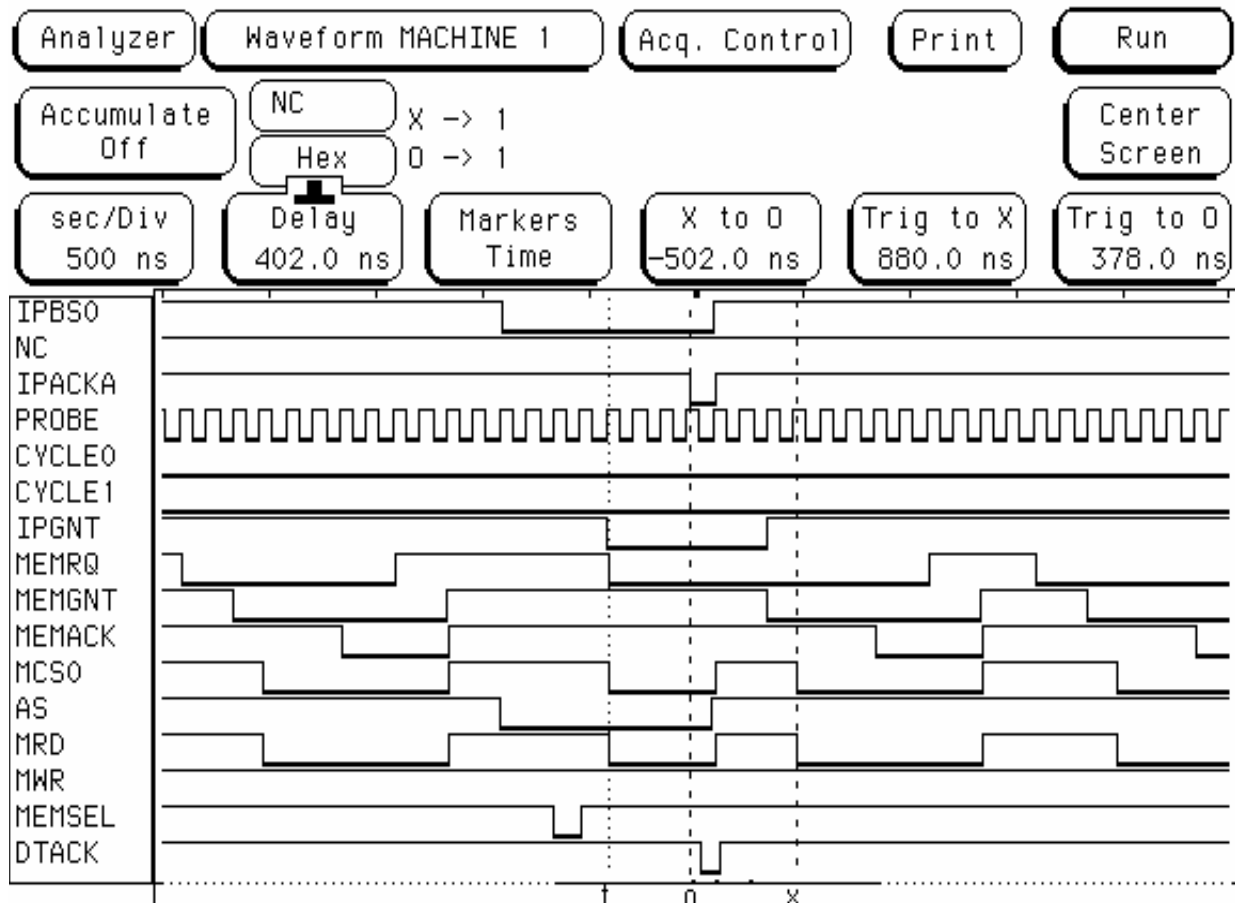
Note: Older version was keeping the signal active until the end of the cycle . When used as a Clock Enable it was latching data at every clock. PowerPC host pipelines address as soon they receive a DTACK on the VME bus. A write to the I/O has the effect to write at two consecutive locations.





12 Additional timing

Some additional arbitration timing between IPBUS and 10k50E



EXAMPLE # 1:

ALPHI TECHNOLOGY CORP.

page 33

REV 2.5

Part Number : 805-25-001-4000

Copyright ALPHI Technology Corporation ,2005

5/16/2005

ARBITRATION BETWEEN IP BUS MEMORY ACCESS(READ) AND 10K50 MEMORY ACCES(READ).

MEMRQ = 10KMEMRQ

MEMGNT= 10KMEMGNT

MEMACK =10KMEMACK.

THE IPBUS HAS CONTROL FIRST OF THE DPR THEN THE 10K50.

10KMEMRQ MUST STAY LOW UNTIL THE END OF THE CYCLE.

10KMEMGNT IS REMOVED TWO CLOCK LATER DUE TO THE 10KMEMRQ FROM THE 10K50 WHICH GET RESYNCHRONIZED WITH THE IPCLK (8MHZ) VME162 WITH VME4SIP

Note :

10KMEMACK IS GENERATED BUT NOT USED INTO ARBITRATION

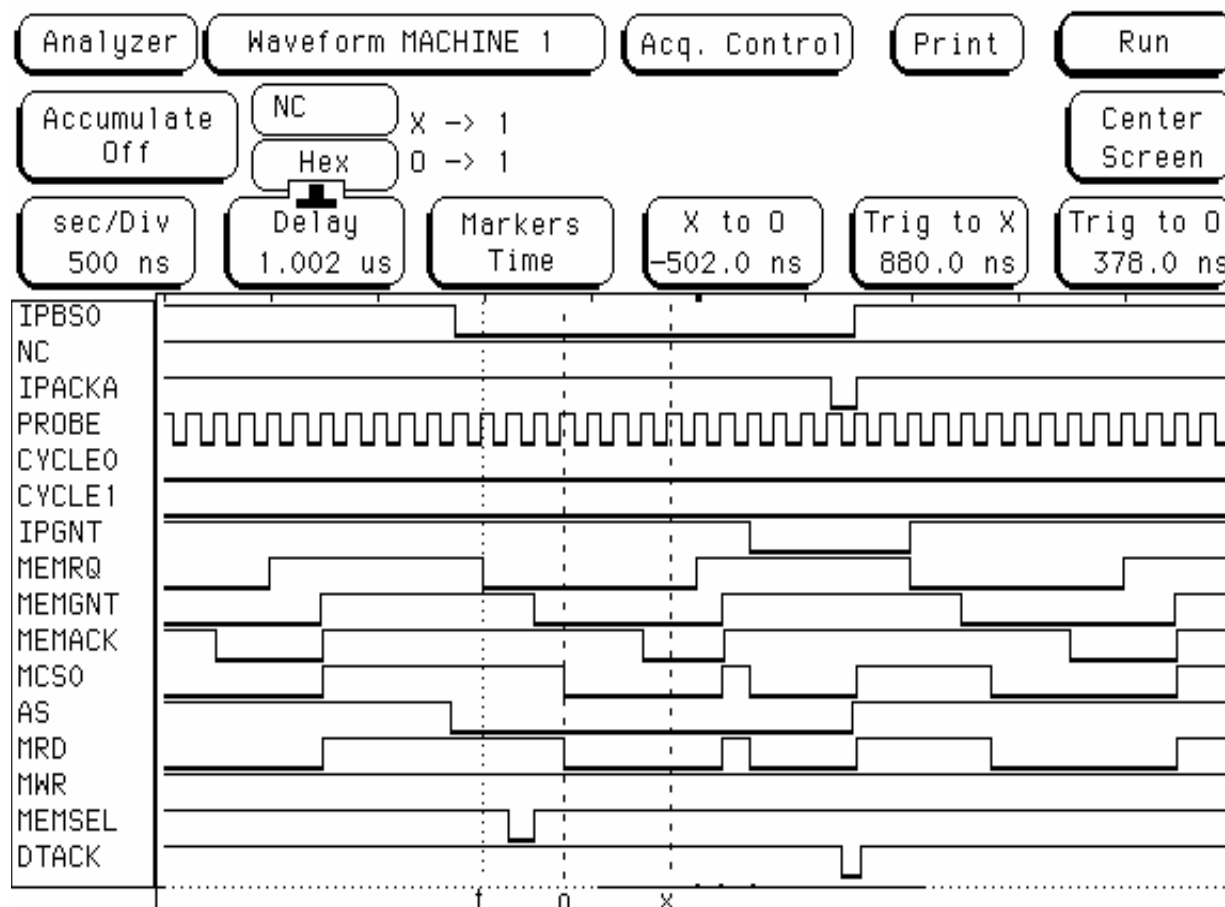
ALL THE DPR SIGNALS ARE SWITCHED INSIDE THE 7160 CPLD.

ALL THESE TEST USE A STATE MACHINE IMPLEMENTED AS EXAMPLE INTO THE 10K50 WITH A 10KMEMRQ GENERATE BY THE GCLK1 CLOCK DIVIDED BY 8.

THE TIMING ACCESS OF THE DPR MATCH THE REQUIREMENT OF 70 ns

nS MEMORY ACCESS. USING GCLK1 AT 8MHZ SHOULD BE NO PROBLEM

Date : 03-04-05



EXAMPLE # 2:

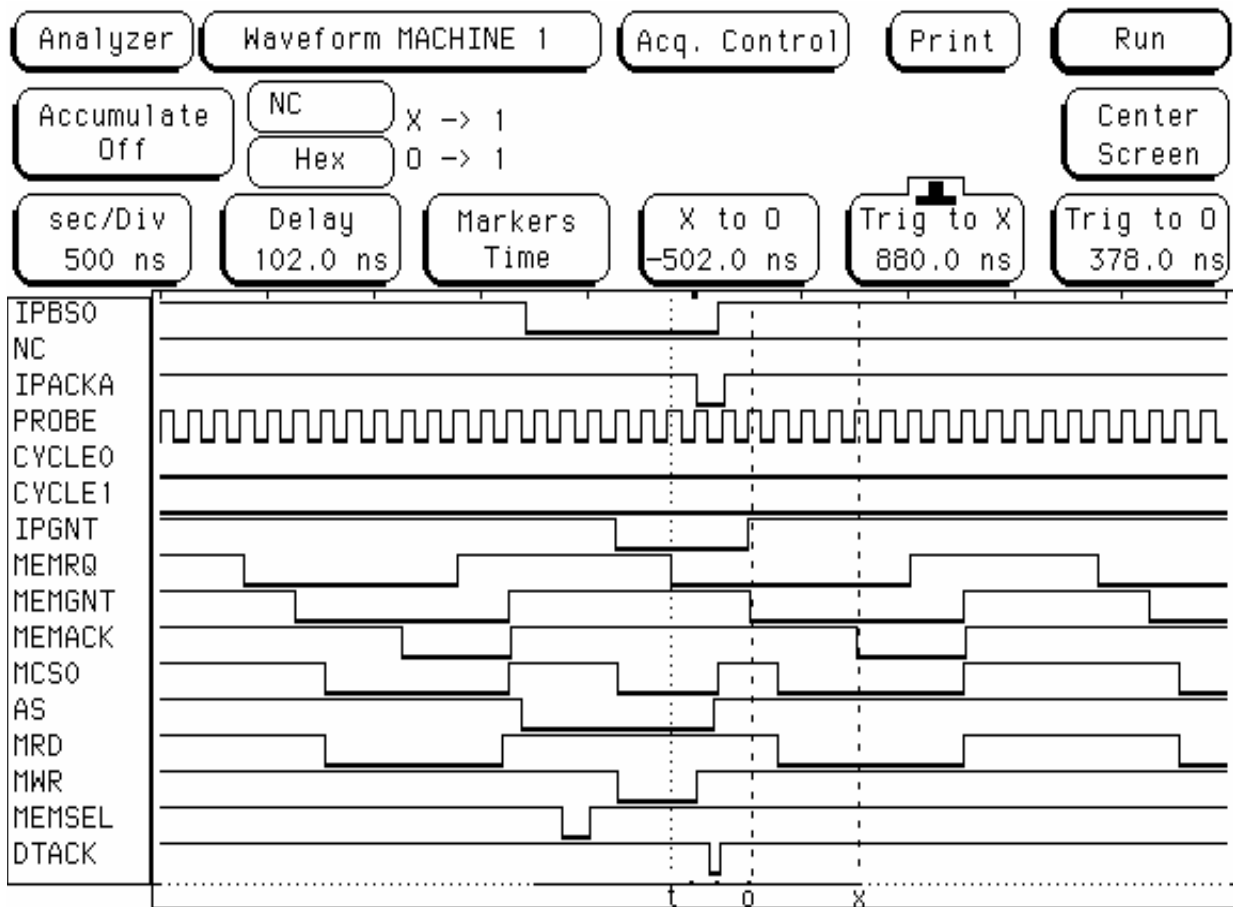
ARBITRATION BETWEEN IP BUS MEMORY ACCESS(READ) AND 10K50 MEMORY ACCES(READ).

MEMRQ = 10KMEMRQ

MEMGNT= 10KMEMGNT

MEMACK =10KMEMACK.

SAME AS ABOVE EXCEPT THAT THE 10K50 GET THE BUS BEFORE THE IPBUS.



EXAMPLE # 3:
 ARBITRATION BETWEEN IP BUS MEMORY ACCESS(READ) AND 10K50 MEMORY ACCES(READ).
 MEMRQ = 10KMEMRQ
 MEMGNT= 10KMEMGNT
 MEMACK =10KMEMACK.

SAME AS EXAMPLE # 1 BUT THE IPBUS GET CONTROL FIRST OF THE DPR THEN THE 10K50. A WRITE IS PERFORMED BY THE IPBUS