VME Commands

VME Address:

Bit 31..25	=	8-bit VME board address = set by dip switches on the board
Bit 24..9	=	don’t care
Bit 8=0	=>	Execute one of the VME commands listed in table 1 Bit 7..4 define the type of the command
Bit 8=1	=>	Send a control sequence to the chip/module to be tested Bit 7..4 define the sequence (table 2). For details on the sequence, consult the ABC(D) specifications document.
Bit 3..0	=	don’t care

Table 1: VME Commands for Board Operation

hex	Read/Write	Description	VME Data Bus
0	Read	Read FPGA Status Register	Bit 15..0=xFACE Bit 16=BUSY Bit 31..17=X
1	Write	Reset Test Vector Memory Counter to 0	X
2	Write	Reset Simulation Vector Memory Cnt to 0	X
3	Write	Clear Histogram Memory	X
4	Read	Read from Histogram Memory and Increment Memory Pointer	Bit 15..0=Stream B Bit 31..16=Stream A
5	Write	Set Base Address for Histogramming and Reset Histogram Memory Counter to xaa000000, where “aa” stands for the Base Address	Bit 7..0=Base Address Bit 31..8=X
6	Write	Send Test Vector	
7	Write	Write to Test Vector Memory and Increment the Test Vector Memory Counter	Bit 17..0=data Bit 31..18=X
8	Write	Write to Simuation Vector Memory and Increment the Simulation Vector Memory Counter	Bit 17..0=data Bit 31..18=X
9	Write	Reset the ReSync FIFO	X
a	Write	Set DAC	Bit 9..0=data Bit 22..19=dac address Bit 18..16=dac number other bits: X
b	Write	Send Convert Signal to ADC	not implemented
c	Read	Read ADC Data from last Conversion	not implemented
d	Read	Read from ReSync FIFO	Bit 16..0=data Bit 17=FIFO empty flag:

			0=empty, 1=not empty Bit 31..18=X
e	Write	Start sending triggers and decode and histogram the data	X
f	Write	Set Frequency	Bit 8..0 = M Bit 10..9 = N Bit 13..11 = T

Table 2: VME Commands used to issue control sequences to the chip or module to be tested.

hex	Read/Write	Description	VME Data Bus
10	Write	Set Trigger-To-Trigger Delay	bit 15..0=delay in cc bit 31..16=X
11	Write	Set Number of Triggers to be sent per Burst	bit 15..0=#triggers bit 31..16=X
12	Write	Send Soft Reset (all chips)	X
13	Write	Send BC Reset (all chips)	X
14	Write	Write to Configuration Register	bit 21..16=chip address bit 15..0=config reg data bit 31..22=X
15	Write	Reset FPGA-Mask Register Pointer	X
16	Write	Write 32 bits to FPGA-Mask Register	31..0=mask reg data
17	Write	Send FPGA-Mask Register to a chip	bit 21..16=chip address bit 15..0=X bit 31..16=X
18	Write	Load Strobe Delay Register	bit 15..0=reg data bit 21..16=chip address bit 31..22=X
19	Write	Load Threshold/Cal DAC Ampl Reg	bit 15..0=reg data bit 21..16=chip address bit 31..22=X
1a	Write	Enable Data Taking	bit 21..16=chip address bit 15..0=X bit 31..16=X
1b	Write	Write Select Register	bit 0: abc(d) select bit bit 1: strobe enable bit
1c	Write	Issue Hard Reset	not implemented
1d	Write	Load Bias DAC	bit 15..0=reg data bit 21..16=chip address bit 31..22=X
1e	Write	Load Trim DAC	bit 15..0=reg data bit 21..16=chip address bit 31..22=X
1f	Write	Set Strobe-To-Trigger Delay	bit 7..0=delay in cc Bit 31..8=X

How the VME Interface Works

1) After startup of the FPGA, it loads the mask registers and the compare registers of the CY964's. The mask registers are all set to 0, and the compare registers are set to:

31..24 = board address (DIP switches)
24..0 = 0

This involves exercising the pins LDS and STROBE* during startup. MWB* is always set to 1. LDS is later on connected to the LDS pin of the 960. For the timing, see p 4-8 of the databook. Writing to the compare reg clears the mask reg. A 0 in the mask reg means that the bit is used for comparison.

2) The 964's respond by asserting VCOMP* if they're hit by an address which matches the compare register.

3) the VCOMP*'s go to the FPGA. The FPGA outputs SVIC_REGION(3 downto 0) are just the inverted VOMP*(3 downto 0).

4) Since we set the board address on bits 31..24 which corresponds to vcomp*(3), only region "1000"=8 is enabled in the setup for the CY960.

There is a setup program, called WinSvic, which creates the bitstream for the configuration PROM of the CY960, The Program is available on the CYPRESS Website, and is installed on the PC in the office 50B-6220, in the folder C:\hubert\new_DAQ\winswic.

The CY960 does 3 things:

a) It creates a programmable chip-select pattern cs(5..0) from the region(3..0) inputs.
b) For each of the 16 possible region-codes, it can be told which types of VME transfers are allowed.
c) Corresponding to the type of transfer, the data byte enable DBE(3:0) signals are asserted. The DBE signals come later than the CS.

So, in case of a VME access to the board address set with the DIP switches, the cs(0) will go high and then also some of the DBE, depending on the access. In case of a 32 bit transfer, all DBE's should go high.
I only check for a coincidence of at least one CS and at least one DBE for 2 clock cycles, and then latch data and address and R_W into the FPGA. The 960 then waits for LACK* to go low to acknowledge the data transfer. So in case of a write access, the LACK* is immediately asserted, in case of a read access, it is asserted when the valid data is on the bus.

At the moment, the 960 is configured in the following way:

i) Initialization: Serial PROM Method (on power up, it loads its config data from PROM)
ii) Configuration=IO
iii) IO Menu:

Only for REGION=8 , the chip select cs(0) is activated. It's set to active HIGH.

The DBE assert time is set to 18 clock cycles. (It's the minimum width in case of self-timed acces. Since we hand-shake the data transfer, it's not so important but must certainly be long enough to be detected by the FPGA).

a) AM Code Menu: Only for region 8: all access modes allowed, all other regions: no access allowed
   b) None of the special modes are allowed. (No lock etc).

Miscellaneous control:

a) decode delay: Thats the time in clock cycles which the 960 waits upon receipt of a AS* on the VME-bus (that's the signal on the VME bus which tells the 960 that a valid address is available) before sampling the region inputs. We set it to the maximum of 5 cc. (That's 100 ns at 80 MHz..).
b) DBE Polarity. I don't use the convention of the databook: I use DBE=active HIGH.
c) AM Code LA bit off. See p 3-48
d) IRQ level=2. Irrelevant since we don't use interrupts.

   e) Bus holdoff=off, see p 3-51.

   f) IACK LACK response = off. See p 3-62

   g) Master interlave=off (only 961?) See online help in winsvic!

The configration file is called **modules.sv**

FPGA pins related to the vme interface:

Signal name	**FPGA In/Out**	**Active**	**Description**
laddr(31:1)	in		local address. Bits 31..24 match the board address for a valid access.
ldata(31:0)	in/out		32 bits of data. Bidirectional.
vme_xcvr_mwb_n	out	low	tie high. Not actively used.
vme_xcvr_lds	out	high	lds signal for the 964. Connected to svic_lds during normal operation. During startup, used to select mask/compare register for loading those.
vme_xcvr_strobe_n	out	low	used to load mask/compare reg on 964 during startup
board_addr(7:0)	in		board address, set by 8 DIP switches

svic_region(3:0)	out	high*	Since only region 8 is enabled in the 960, svic_region(3)=not vcomp(3). svic_region(2 downto 0)=”000”
svic_lack_n	out	low	used to acknowledge local data transfer. Immediately asserted after write access, asserted after valid data has been put on ldata bus after a read access.
svic_lds	in	high	must be tied to vme_xcvr_lds during normal operation.
svic_lirq	out	low	must be high always. Not used.
vcomp(3:0)	input	low	result of address comparison in the Cy964’s. Only vcomp(3) is used.
svic_dbe(3:0)	in	high*	these bits go high during a valid vme access corresponding to the type of vme access (32 bit, 16 bit etc).
svic_cs(5:0)	in	high*	only cs(0) is enabled. See description above.
svic_lden_n	in	low	not used.
svic_pren_n	in	low	asserted during initialization. Not used.
svic_swden_n	in	low	swap data enable. Not used.
svic_r_w*	in	--	write=low, read=high
svic_strobe	in	high	only used if 960 configures 964 on power up. Not used in this design.

- *the polarity is programmable in the CY960 during configuration*