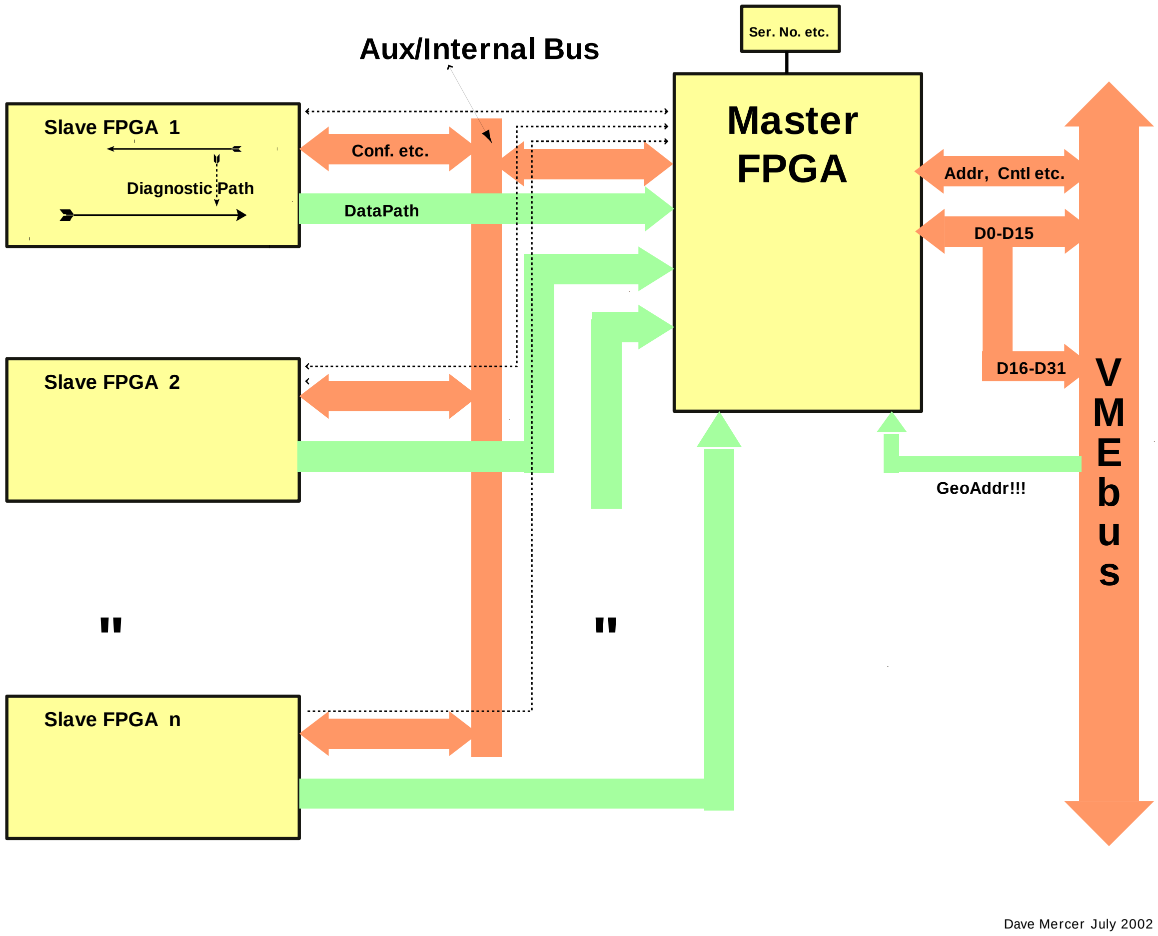

Fig 1

To be read in conjunction with Paul's <Specification.ps>; in particular my Fig 1 is to be compared with Fig 5 in the Spec.

NOTES (in no particular order)

- Geographical Addressing.

We have a VME crate full of Readout Modules and they have to be individually addressed. The old barbaric way was to use hex switches on every board making sure they were all different…..impossible on a night shift. We will use 5 spare pins on the VME J2 connector with a hard wired address which the module can "read"; pull ups are on the module and backplane pins are grounded or not. Access to the rear of the motherboard is needed but only once. The 2 outer rows of the J2 96 way connector
are usually left long and can be wire wrapped to an adjacent gnd pin. There is a third method using the Bus Request and Grant daisychain to do an autoconfigure but I did it such a long time ago I cannot for the life of me remember how clever I used to be !!

- Aux/Internal bus (16 bit data; xxx bit address)

This is for configuration and diagnostics ; the fast datapath is separate .. see Fig 1. It is a very *simple* synchronous bus because all FPGAs share a (carefully routed) common clock. The VME FPGA is always master and the Slave FPGAs always slaves !

ie the slaves only speak when spoken to ; therefore there is never any contention . For any eventuality I will add a bussed LocalBusy signal so that slaves can signal to the master they're busy (collectively busy ,all slaves being equal). This should not be needed in normal running.

The VME synchronization and handshaking is done in the VME FPGA
I will generate the exact specification including timing ASAP

- Sideband Signals

These are shown dotted above and are point to point , say about 4

1. Master to Slave ; <slave FPGA addressed >. The sub-address decoding to address the Slave FPGA's is done in the Master FPGA . This way all the slave FPGAs can have *IDENTICAL* VHDL inside. Indeed they could even be configured (loaded) in parallel saving money and time….but this needs further discussion perhaps.

2. Slave to Master ; < slave FPGA request attention. > for any reason error conditions etc.

3 , 4 spare

Who said CAMAC is dead!!

- Datapath (16 bit internally / 32 bit to VME)

A24/D32 , unidirectional (Read Only) using VME Block Mode
N.B. there must be a separate path from the Aux/Internal space to the Datapath to facilitate commissioning /troubleshooting. I suggest BOTH in the Slave and Master FPGAs separately

- All on board devices inc. FPGA's 3v3 I/O except possibly VME Buffer chips .

FPGA internal supply 1.8 or 2.5v depending on family chosen.
(I must check for full VME spec. compliance for XILINX … I've already been there for Altera) It may be that the Xilinx has not got enough 'ummph' to drive the VME directly,
in which case 74ABT(E) buffers will be used (not shown above). It might be more versatile to have D16->31 direct from FPGA ,not as shown .

- All internal registers must be Read / Write ; ie nothing read only.

- Master FPGA has unique , hard wired, serial number eg. Dallas "one wire" plus optional EEPROM I2C or "one wire"

- Master FPGA is VME interrupter ;
  slave FPGAs interrupts via Master (via sideband signal)

- Diagnostic 4bit Hex/Dil switch to FPGA

- Diagnostic LEDS on front panel (Red => error, otherwise green)

- Logic analyzer header connected to FPGA(s)

- JTAG output from Master to (re)program Slaves
  Initially via software from VME (like JAM player or bit bashing)
  Later hardware assisted if necessary.

- Load all FPGA's in one chain from a single "big" Configuration SROM; perhaps all slave FPGA's loaded in parallel ?

- Use XILINX FPGA's ? Spartan IIe

I'm not yet clear how the datapath from slave to master works!!

My simplest scenario is as follows :-

Each event data block is of a fixed length (I mean fixed from event to event and also the same for every slave , fixed but programmable)

All slaves transmit their blocks at the same time , in parallel , to the master.

Synchronizing the start of the block could be with another sideband signal.

The master's data input looks like a FIFO and if all slaves send in parallel then only one set of internal address pointers are needed in the master FPGA , otherwise ….

To the VME all data appear simply memory mapped and each slave's data being contiguous with it's neighbour

I need to get my head round the exact numbers involved to make sure the internal RAM will be sufficient .This , together with the total pin count, will determine the exact FPGA (both footprint and the logic size) we choose.

Another even simpler :-

Suppose that the data is left in the Slave FPGAs and readout via the master in one big DMA (same maximum VME speed as above) the data are then readout sequentially from the Slaves… So point to point datapaths are NOT necessary ,a 16 bit bus will do ….. But we've got one already!!!! The datapath and control functions are not incompatible and can be easily made to mutually co-exist without interference with careful design. The down side is that the VME readout deadtime is now 1st order; compared to above being 2^{nd} order.. Any comments ???

NOTE .

I have only considered the VME interfacing aspect so far.

Dave Mercer Saturday, August 03, 2002