HUNT ENGINEERING
Chestnut Court, Burton Row,
Brent Knoll, Somerset, TA9 4BP, UK
Tel: (+44) (0)1278 760188,
Fax: (+44) (0)1278 760199,
Email: sales@hunteng co.uk
http://www.hunteng.co.uk
http://www.hunt-dsp.com

HUNT ENGINEERING

HERON2

HERON Processing Module for C6203

USER MANUAL

Hardware Rev A
Document Rev A
P.Warnes 15/7/2002

COPYRIGHT

WARRANTIES LIABILITY and INDEMNITIES

HUNT ENGINEERING warrants the hardware to be free from defects in the material and workmanship for 12 months from the date of purchase. Product returned under the terms of the warranty must be returned carriage paid to the main offices of HUNT ENGINEERING situated at BRENT KNOLL Somerset UK, the product will be repaired or replaced at the discretion of HUNT ENGINEERING.

Exclusions - If HUNT ENGINEERING decides that there is any evidence of electrical or mechanical abuse to the hardware, then the customer shall have no recourse to HUNT ENGINEERING or its agents. In such circumstances HUNT ENGINEERING may at its discretion offer to repair the hardware and charge for that repair.

Limitations of Liability - HUNT ENGINEERING makes no warranty as to the fitness of the product for any particular purpose. In no event shall HUNT ENGINEERING'S liability related to the product exceed the purchase fee actually paid by you for the product. Neither HUNT ENGINEERING nor its suppliers shall in any event be liable for any indirect, consequential or financial damages caused by the delivery, use or performance of this product.

Because some states do not allow the exclusion or limitation of incidental or consequential damages or limitation on how long an implied warranty lasts, the above limitations may not apply to you.

TECHNICAL SUPPORT

Technical support for HUNT ENGINEERING products should first be obtained from the comprehensive Support section www.hunteng.co.uk/support/support.htm on the HUNT ENGINEERING web site. This includes FAQs, latest product, software and documentation updates etc. Or contact your local supplier - if you are unsure of details please refer to www.hunteng.co.uk for the list of current re-sellers.

HUNT ENGINEERING technical support can be contacted by emailing support@hunteng.co.uk, calling the direct support telephone number +44 (0)1278 760775, or by calling the general number +44 (0)1278 760188 and choosing the technical support option.

TABLE OF CONTENTS

The HERON module is a module defined by HUNT ENGINEERING to address the needs of our customers for real time DSP systems. The HERON module is defined both mechanically and electrically by a separate HERON module specification that is available from the HUNT ENGINEERING CD, via the documentation section from the CD browser, or online from http://www.hunteng.co.uk .

The HERON module specification also defines the features that a HERON module carrier must provide. HERON stands for Hunt Engineering ResOurce Node, which tries to make it clear that the module is not for a particular processor, or I/O task, but is intended to be a module definition that allows "nodes" in a system to be interconnected and controlled whatever their function. In this respect it is not like the TIM-40 specification which was specific to the 'C4x DSP.

As the HERON2 was developed, HUNT ENGINEERING have developed HERON modules carriers like the HEPC9, other HERON processor modules (that carry various other members of the TMS320C6000 family of DSP processors from TI), and new HERON-IO and HERON-FPGA modules. In addition to these modules, the HERON specification is a super-set of the pre-existing HUNT ENGINEERING GDIO module, so the GDIO modules from our C4x product range can also be used in HERON systems.

The HERON module connects to the carrier board through several standard interfaces.

- The first is a FIFO input interface, and a FIFO output interface. This is to be used for the main inter-node communications. (It is also used for connection to the HOST computer, if any).
- The second is an asynchronous interface that allows registers etc to be configured from a HERON module. This is intended for configuring communication systems, or perhaps to control some function specific peripherals on the carrier board.
- The third is a JTAG (IEEE 1149.1) interface for running processor debug tools.
- The fourth is the Heron Serial Bus interface which may also be used for configuring communication systems, function specific peripherals on the carrier board, and also I/O modules.
- The last is the general control such as reset, power etc.

HUNT ENGINEERING defined the HERON modules in conjunction with HEART – the Hunt Engineering Architecture using Ring Technology. This is a common architecture that we will adopt for our HERON carriers that provides good real time features such as low latency and high bandwidth, along with software reconfigurability of the communication system, multicast, multiple board support etc., etc.

However, it is not a requirement of a HERON module carrier that it implements such features. In fact our customers could develop their own module carrier and add our HERON modules to it. Conversely our customers could develop application specific HERON modules themselves and add them to our systems.

The HERON2 is HERON processor module that carries the TMS320C6203 (300MHz fixed point processor giving 2400 MIPS).

At this time the most popular HERON module carrier that is being shipped is the HEPC9.

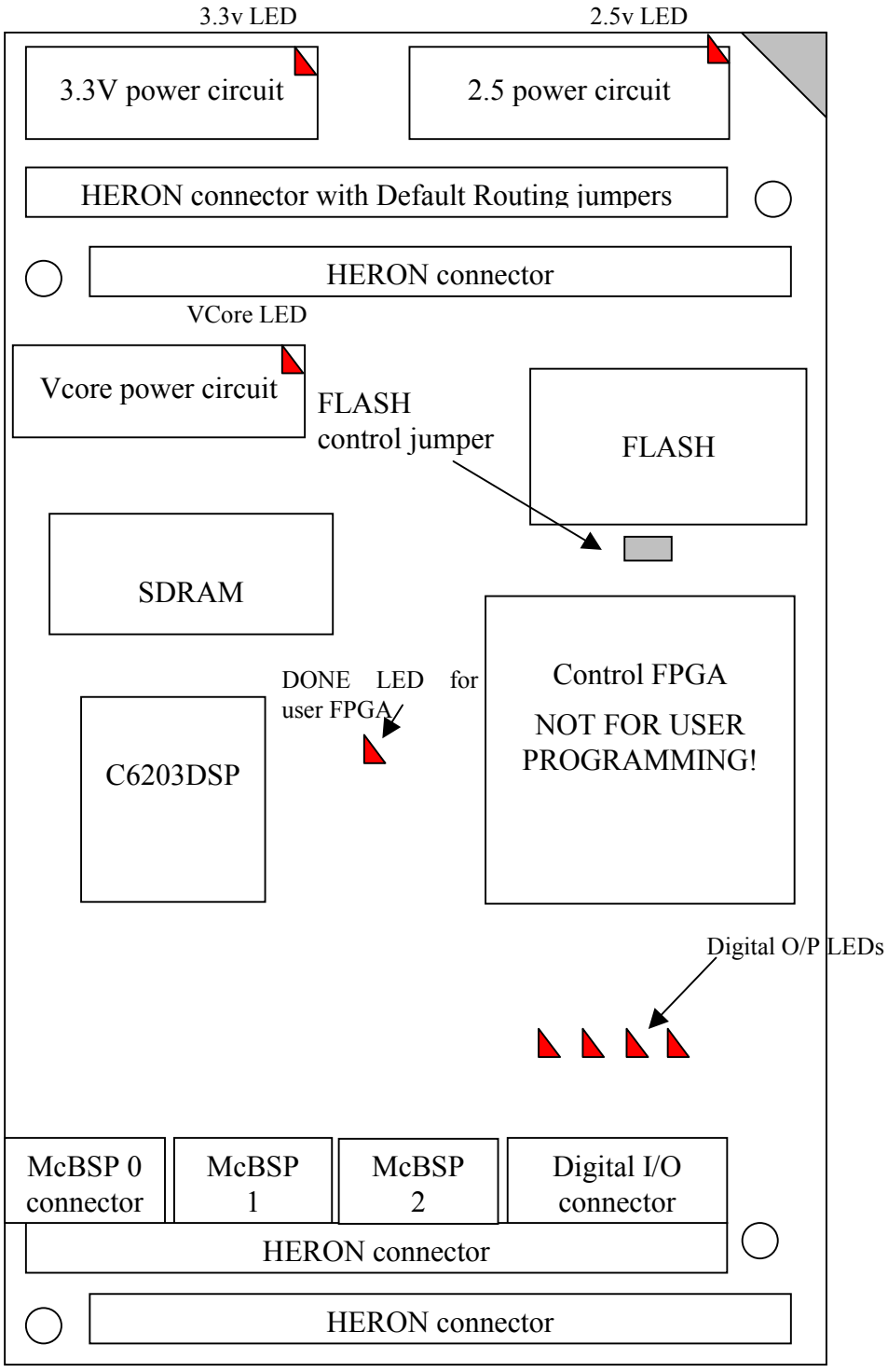
HERON2s use a FIFO clock of 75Mhz which is compatible with the HEPC9, but is actually not compatible with the older module carrier HEPC8.

The HERON2 provides the processor along with 32Mbyte of Synchronous DRAM (SDRAM).

The HERON2 provides access to the three Multi-Channel Buffered serial ports (McBSP) of the processor, via connectors on top of the HERON module. However, one of these, McBSP#2, is also used by the HSB and UMI features of the HERON2. This means that the McBSP2 can be used for **either** HSB or UMI interrupt functions **or** as a McBSP.

The HERON2 provides 4 bits each of Digital Inputs and Digital Outputs, via a connector on top of the HERON module.

The HERON2 pre-boots from a FLASH ROM on the module, in order to start booting from the HERON FIFO #0. The remainder of this FLASH ROM is available for the user to store non-volatile variables, or in the case of an embedded system to store the entire boot sequence.


3.3v LED
2.5v LED
3.3V power circuit
2.5 power circuit
HERON connector with Default Routing jumpers
HERON connector
VCore LED
Vcore power circuit
FLASH
control jumper
FLASH
SDRAM
DONE LED for
user FPGA
Control FPGA
NOT FOR USER
PROGRAMMING!
C6203DSP
Digital O/P LEDs
McBSP 0
connector
McBSP
1
McBSP
2
Digital I/O
connector
HERON connector
HERON connector

The HERON2 is a module that plugs into a HERON module carrier.

The HERON2 should be fitted to the carrier card along with any other modules that your system has and their retaining nuts fitted (see a later section of this manual for details).

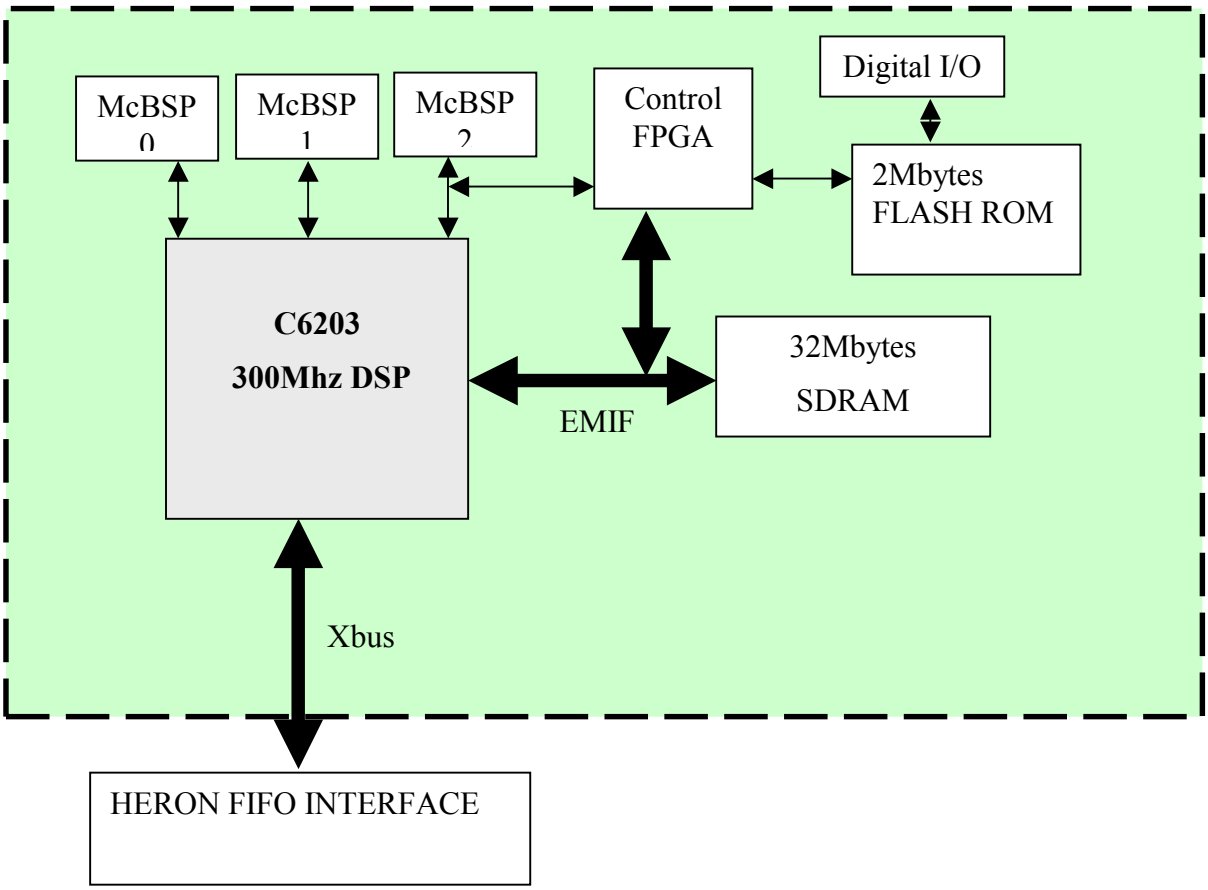
There is only one user configurable jumper on the HERON2, used as a Write protect for the FLASH ROM. Most applications will not need to change it from its default (shipping) condition which is "fitted".

The HERON2, following reset, will automatically boot the C6203 DSP with a pre-boot program. This program performs all of the processor initialisation to be able to correctly use the peripherals provided on the module. This means that the user program should not need to configure hardware such as memory control registers.

Following the initialisations, the pre-boot code will start to boot from the data provided via HERON FIFO #0. This means that it will look at the flags for that FIFO, and when data arrives it will read it from the FIFO. The format of that boot stream is defined later in this manual, but essentially it first receives a "carrier ID" and a "module ID". While this combination does not match the carrier and module IDs of this module, the boot stream is read and discarded. When there is a match, it reads and stores the boot data and then jumps to that boot data.

This provides a means for code to be downloaded to the processor via the HERON FIFOs by a tool such as the HUNT ENGINEERING Server/Loader.

Software that runs on the HERON2 will be written in C or C callable assembler, compiled with the TI Code Generation Tools which are part of Code Composer Studio. HUNT ENGINEERING provides a CCS Plug in that can make your project for you ensuring that it is correctly configured for the HERON2. HUNT ENGINEERING provides the HERON-API software to allow you to access the hardware features through a consistent interface library. This is very important as other HERON processor modules will provide the same features, but due to differences in processor type or hardware design will probably be accessed in a different way. Using the HERON-API means that you can simply re-compile your DSP application for use on another HERON processor module type without changing the code.



The main data paths are shown, control paths are omitted for clarity.

The memory map of the processor is discussed here for information only. Using the "Create New HERON project" plug in supplied by HUNT ENGINEERING for Code Composer Studio in conjunction with HERON-API means that your program is automatically configured to use the correct addresses.

Address	C6203 function	HERON2 function
0 → 0x3ffff	Internal Program RAM 0	Program memory
0x40000 → 0x5ffff	Internal Program RAM 1	Program memory /Instruction cache
0x6 0000 → 0x3f ffff	Reserved	
0x40 0000 →0x13f ffff	External Memory Interface (EMIF) CE0	On module peripherals controlled by "control" FPGA
0x140 0000 → 0x17f ffff	External Memory Interface (EMIF) CE1	FLASH memory and off module Peripheral access
0x180 0000 → 0x1ff ffff	Internal Peripheral Bus various control registers	
0x200 0000 → 0x2ff ffff	External Memory Interface (EMIF) CE2	16Mbytes SDRAM
0x300 0000 → 0x3ff ffff	External Memory Interface (EMIF) CE3	16Mbytes SDRAM
0x400 0000 → 0x3fff ffff	Reserved	
0x4000 0000 → 0x4fff ffff	Xbus CE0	HERON FIFOS
0x5000 0000 → 0x5fff ffff	Xbus CE1	Not Used
0x6000 0000 → 0x6fff ffff	Xbus CE2	Not Used
0x7000 0000 → 0x7fff ffff	Xbus CE3	Not Used
0x8000 0000 → 0x8007 ffff	Internal Data RAM	Internal Data RAM
0x8008 0000 → 0xffff ffff	Reserved	

SDRAM

The HERON2 has 16Mbytes of SDRAM connected to its CE space 2, configured as 4M 32bit words. It has a further 16Mbytes of SDRAM connected to its CE space 3, configured as another 4M 32bit words.

This results in a single memory area of 32Mbytes.

The hardware initialisation code executed from the FLASH ROM correctly initialises CE2 and CE3 controls to access this memory as follows:-

emif_ce2_timing = 0x00000030 and

emif_ce3_timing = 0x00000030 and

emif_sdram_control = 0x07229000 and

emif_sdram_refresh = 0x0000093f.

The user can access the SDRAM memory at address 0x02000000 → 0x04000000.

The SDRAM is accessed in a single cycle of the C6203 External Memory Interface (EMIF) with a pipeline delay of 3 cycles. See the separate document on external memory types for the C6000, on the HUNT ENGINEERING CD or web site.

The SDRAM on the HERON2 is rated at a clock speed of 166MHz, but SDRAM is always accessed by the C6203 at half of the processor clock, which is 150Mhz.

HERON FIFOS

The HERON FIFOS (provided by the HERON module Carrier card) can be accessed by the C6203 in CE space 0 of the Xbus interface.

The modes of the Xbus are correctly initialised by pull ups/downs on the Xbus data bus. The remaining Xbus configuration is made by the boot loader program that executes from FLASH ROM following a reset. The registers are set as follows:-

Xbus Global Control Register = 0x6000

The Xbus cannot be accessed by processor cycles, but instead **must** be accessed using a DMA engine.

The FIFOs are addressed on the Xbus as follows:-

Read FIFO#0	0x4000 0000
Write FIFO#0	0x4000 0000
Read FIFO#1	0x4000 0004
Write FIFO#1	0x4000 0004
Read FIFO#2	0x4000 0008
Write FIFO#2	0x4000 0008
Read FIFO#3	0x4000 000C
Write FIFO#3	0x4000 000C
Read FIFO#4	0x4000 0010
Write FIFO#4	0x4000 0010
Read FIFO#5	0x4000 0014

Write FIFO#5 0x4000 0014

The FIFOs are clocked by the processor clock divided by 4, i.e. 75Mhz. N.B. This makes it impossible to use the HERON2 on an HEPC8 module carrier card.
Note that the FIFO clock does not run until the Xbus Global Control Register is written. This register is correctly initialised by the pre-boot loader that runs from the FLASH ROM, but the loading from FLASH ROM takes around 4.5ms. After that clock starts to run there may be some further delay while the HEART devices lock to that clock. For this reason the HERON FIFOs should not be accessed for 10ms after a reset. Of course any program running from the FLASH memory will automatically have the correct delay as it is loaded (by the pre-boot loader) after the register has been set. It is important though to make certain that nothing tries to send data to the HERON2 during that time as this can cause the HERON FIFO (that is not being clocked properly) to lock up. Thus booting utilities like the HUNT ENGINEERING Server/Loader must introduce this delay, and I/O devices should take notice of the Config signal before starting to send data.
There is also a register, the Flags Register, where the FIFO flags can be read. This may be used to determine when single-word or block accesses to the FIFOs may proceed. Details of the HERON FIFOs and the Flags Register can be found in the appendix, but it is strongly recommended that all FIFO accesses be made using the HERON-API software. See the later Software section in this manual.
The Flags register is addressed in EMIF CE0 memory space – see the EMIF CE0 peripherals section.
The HERON2 allows DMA accesses of the FIFOs to be synchronised to the relevant FIFO flags. This enables the DMA burst accesses to start and stop, without CPU intervention. See appendix for details on DMA synchronisation.

EMIF CE1 peripherals

The HERON2 uses CE space 1 to address 4Mbytes of FLASH ROM, and also the "off module" peripherals as defined in the HERON Module specification. All accesses are made as asynchronous peripherals.

At reset the C6203 initialises the memory access time to be their maximum values, allowing access of (relatively slow) FLASH memory before initialising any EMIF control registers.

FLASH ROM

The HERON2 has 2Mbytes of FLASH ROM connected to its CE space 1, configured as 512k x 32.

The user can access this memory at address 0x140 0000 → 15f ffff, when the CE1 control register is set correctly for FLASH accesses.

The FLASH ROM used on the HERON2 has a software write protect, which needs to be overridden before the ROM can be written to from software. There is a FLASH ROM programming utility provided on the HUNT ENGINEERING CD, which includes this software "unlock".

The HERON2 has a jumper setting that write protects the FLASH ROM. It is this FLASH ROM that the 'C6203 boots from.

See the later section on FLASH ROM control jumper.

The actual FLASH ROM fitted to HERON2s can vary in manufacturers and can also be "top boot block" or "bottom boot block". The manufacturer and type can be identified by reading registers from the FLASH parts. The software that HUNT ENGINEERING provide is able to use that to detect the correct type of programming to use.

The FLASH ROM is slow and can be accessed with the processor default settings (that is sets following a reset) but its needs actually are :-

Read Setup	3 cycles
Read Strobe	40 cycles
Read hold	4 cycles
Write Setup	3 cycles
Write Strobe	40 cycles
Write hold	4 cycles

This can be achieved with a CE1 control register value of 0x4A342823 giving accesses of 47 (300mhz) cycles. The C6000 silicon can only generate cycles this fast if using DMA. However the HERON2 hardware will automatically extend the read and write strobe times to achieve this if the control register is set to use less cycles than this. This can be useful if FLASH ROM accesses are used at the same time as Off module Peripheral cycles which can (according to the application) be shorter.

Off Module Peripherals

The HERON2 provides the facility to access off-module hardware that may exist on the carrier board into which the module is plugged. This feature may be used, for example, for programming configuration hardware on the carrier board, or for accessing any shared memory that may be available.

The HERON2 makes these accesses using the CE1 memory space that can be configured to provide the correct cycle length to suit the speed of the hardware being accessed.

Up to 2Mbytes of off-board hardware may be accessed, as 512k x 32-bit words, from addresses 0x01600000 to 0x017FFFFC.

The off module peripherals are not part of the HERON2 design so could require any cycle length. For help in understanding how to access your own peripherals see the appendix on this subject.

For "normal" use of the HERON2, the processor will assume its longest possible cycles following a reset. This will allow it to load the boot code from the FLASH memory. The Boot program will set the CE1 control register to the correct value for further FLASH memory accesses.

If the user then requires access to the Off module peripherals, the CE1 control register must be changed to the required value, which should also allow for accessing of the FLASH ROM because of the use of the ARDY signal to extend FLASH ROM cycles.

EMIF CE0 peripherals

The HERON2 uses CE space 0 to access all of its on module peripherals. All accesses in this memory space are treated as asynchronous accesses.

The On module peripherals need to be accessed as fast as possible, and what is possible

is :-

Read Setup	3 cycles
Read Strobe	4 cycles
Read hold	4 cycles
Write Setup	3 cycles
Write Strobe	4 cycles
Write hold	4 cycles

This is achieved with a CE1 control register value of 0x41340423 giving 11 (300mhz) cycles per access. Actually the C6000 silicon can only generate cycles this fast if using DMA – an unlikely way of accessing such peripherals. Simple CPU accesses will take approximately 20 cycles.

This register is set by the boot loader which is executed from FLASH ROM after reset.

Address	Register
0x40 0000	FIFO Flags register A
0x40 0004	EXTINT4 interrupt enable register
0x40 0008	EXTINT5 interrupt enable register
0x40 000C	EXTINT6 interrupt enable register
0x40 0010	EXTINT7 interrupt enable register
0x40 0014	Digital I/O
0x40 0018	Module/Carrier ID
0x40 001C	Config
0x40 0020	UMI_OUTPUT control
0x40 0024	UMI_TIMER0 control
0x40 0028	UMI_TIMER1 control
0x40 002C	UMI_INPUT
0x40 0030	HSB data
0x40 0034	HSB control/flags
0x40 0038	HSB target address

FIFO Flags

The user can access the flags register at address 0x40 0000, but HERON-API should be used to manage FIFO accesses. For details of the register see the FIFO accessing appendix.

Digital I/O

The HERON4 provides 4 bits of TTL digital input and 4 more bits of TTL digital output. The bottom 4 bits of the DSP word are used to access these I/Os. All other bits during that access are undefined.

The user can access this memory at address 0x40 0014, but using HERON-API functions is recommended to maintain hardware independence.

See the appendix for details of this register

Module and Carrier ID

The HERON specification assigns pins on the HERON module that give a HERON module access to the 4-bit carrier ID of the carrier that it is plugged into, and a unique 4-bit HERON module slot ID. The module's boot program makes use of this, and it is imagined that user programs might also find it useful to be able to uniquely identify themselves in the system, e.g. when printing an error message.

The user can access this memory at address 0x40 0018, but using HERON-API functions is recommended to maintain hardware independence.

Config

The HERON specification defines an open collector Config signal that is driven low by a processor module following reset. Each processor releases the drive from the Config signal when they have booted and are ready to operate.

I/O modules can therefore use the Config signal to halt their operation until the processors in the system are ready.

The HERON2 removes the drive from the Config signal by writing the value 0 to address 0x40 001C. It can re-assert the drive by writing a 1 to the same address. The state of the Config signal can be read from this same address. The bottom bit (bit0) is low when config is asserted in the system.

To maintain hardware independence of your code it is recommended that the Config signal is controlled using the relevant HERON-API functions.

Uncommitted Module Interconnects

There are some "Uncommitted Interconnect" signals defined by the HERON specification, which are simply connected to all modules.

These are intended to connect control signals between modules, for example a processor module can (via software) drive one of these signals with one of is timer outputs. Then if an I/O module can accept its clock input from one of these signals, it is possible to implement a system with a programmable clock. There will be other uses for these signals that are module design dependent.

The HERON2 can drive one of these with a Timer output or digital level. It can also use them to receive an interrupt or as a timer input, or can read their state.

See the appendix for details of how to use these connections, but it is strongly recommended that all accesses are made using the HERON-API software.

Heron Serial Bus Controller

The HERON2 has a Heron Serial bus interface that is accessed in CE0 space.

There are three registers at addresses 0x40 0030, 0x40 0034 and 0x40 0038

For details of the use of this interface see the appendix on this subject.

HERON Module Type

The HERON2 module implements all four of the HERON connectors, which means it is a 32 bit module that can access all twelve of the possible HERON FIFOs.

For a complete description of the HERON interfaces, signal definitions and connector types and pin outs, refer to the separate HERON specification document. This can be found on the HUNT ENGINEERING CD, accessed through the documentation viewer, or from the HUNT ENGINEERING web site at http://www.hunteng.co.uk.

The HERON2 has a processor so asserts the "Module has processor" pin as defined in the HERON specification.

The HERON2 supports JTAG so asserts the "Module has JTAG" pin as defined in the HERON specification.

The HERON2 has a serial bus so asserts the "Module has serial bus" pin as defined in the HERON specification.

The HERON2 has a 32 bit interface so asserts the "Module width has" pin low.

Hardware Reset

Before the HERON2 can be used, it must be reset. This reset initialises all of the HERON2 circuitry and forces the DSP to boot from the FLASH ROM.

This signal is driven by the HERON module Carrier and must NOT be left unconnected, as this will cause the HERON2 to behave erratically.

Software Reset (Code Composer Studio)

Code composer has a menu that allows "DSP reset". This must never be confused with the hardware reset controlled via the HERON pin – it is not the same thing. The Code Composer "DSP reset" simply resets some of the internal registers of the DSP but will NOT perform a hardware reset.

It cannot affect the HERON carrier board FIFOs, or any other module in the system.

Hunt Engineering Reset Plug-in (Code Composer Studio)

Code composer has a tools menu, and when the HUNT ENGINEERING software is installed entries are added to that menu. Under Tools→HUNT ENGINEERING→Reset Plug-in you will be able to get a "System reset" button. Using this reset from within Code Composer Studio is the only reliable way to reset the HERON2. This plug in actually lets the processor "run free", then asserts the hardware reset. This clears all FIFOs, asserts the Config signal and resets the DSPs. Because the DSPs are running free they can boot from ROM and correctly initialise the EMIF control registers. If these registers are not correctly initialised after reset HERON2 hardware will not be correctly accessed by the DSP and will

cause "cannot verify memory at xxxxx" messages from Code Composer Studio. It can also cause FIFO accesses to fail because the flags are not correctly read.

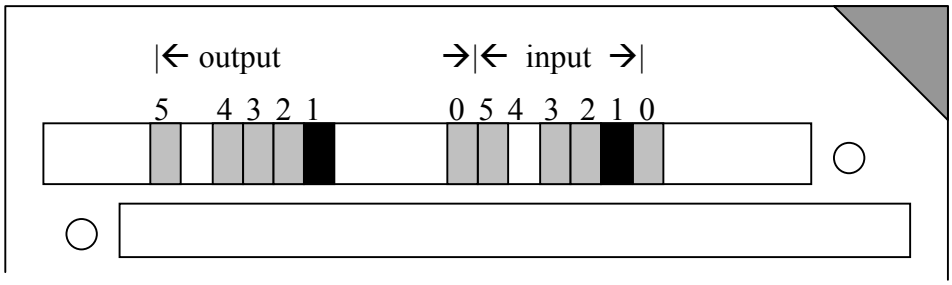
Config

There is a system wide Config signal that is open collector and hence requires a pull up to be provided by the motherboard. The HERON2 drives this signal active (low) following a hardware reset, and the drive must be removed by the application software.

The Config signal is provided for any hardware that needs to be disabled until the entire system has booted. If there is any hardware that uses this feature it will not function until all processors have removed their config signal.

Default Routing Jumpers

The default routing jumpers are provided by HERON modules. These are the longer pins on the topmost HERON connector of the module. These pins protrude above the HERON module when it is fitted to the module carrier, to which jumper links can be fitted.



These jumpers are used to select the default routing of FIFO 0 on the Carrier card. This is to enable the actual source of the boot stream to be set up, even before the application starts and configures the system connectivity to be used.

The exact use of these jumpers is defined by the HERON module carrier, so you should reference the user manual for the Carrier card you are using, but the numbering of these "default routing" jumpers is defined in the HERON spec, and shown above.

For example, the jumpers as fitted in the diagram show that the default routing for both the input FIFO #0 and the output FIFO #0 is selected as 1.

HERON processor modules accept their boot stream from FIFO 0, which is why these jumpers are provided for FIFO #0 only.

The HERON4 will boot from FIFO 0, so the Default Input FIFO 0 needs to be set to the FIFO number that will contain the boot stream. By convention the default output FIFO number will be set to be the same, and higher level software might rely on this being so. The HUNT ENGINEERING server/loader in fact relies on you setting the default output FIFO to be the same as for the input FIFO so that a protocol handshake can take place.

Incorrect setting of these jumpers will cause the system to hang instead of booting.

Physical Dimensions of the Module

A size 1 HERON module is 4.0 inches by 2.5 inches overall.

The 5mm limit on component height under the module is not violated by the HERON2.

The maximum height of the HERON2 above the module is 6.5mm.

This means that the assembly of a HERON module carrier and the HERON2 is less than the 20mm single slot spacing of PCI, cPCI and VME.

Power Requirements of the HERON2

The HERON2 only uses power from the 5V HERON supply. The 3.3V 2.5V and 1.5V required by the FPGA and DSP are generated on board from this +5V.

The maximum rating of the HERON2 is 1.5A, but the typical (and measured) value is nearer 0.6A.

Digital I/O Connector

The Digital I/O connector provides the possibility to control external equipment directly under processor control.

Input characteristics

There are 4 TTL compatible inputs, received on the HERON2 by a Xilinx device.

The Inputs have the following characteristics:

SIGNAL	MIN	MAX
Vih	2.0V	
Vil		0.8V

The input current requirement of the device is very low (10uA) but the inputs are pulled high on the HERON2 using a 10K resistor to +5v. This makes the input impedance 10K.

Output characteristics

There are four TTL compatible outputs, driven by the HERON4 using a Xilinx device.

The Outputs have the following characteristics:

SIGNAL	MIN	MAX
Voh	2.4V	
Vol		0.4V

The output current of the device is as follows:

SIGNAL	MIN	MAX
Ioh	-24mA	
Iol		24mA

There are also LEDs which are driven by a buffered version of these outputs. They illuminate when the digital output is a 0.

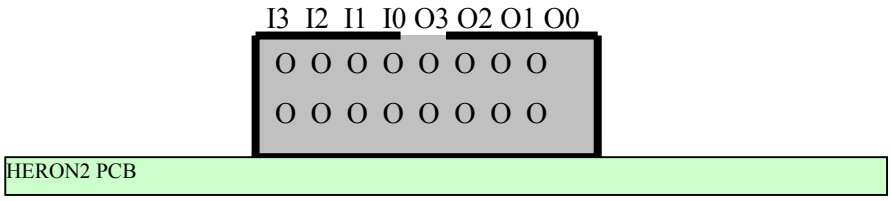
Connector type

The Digital I/O connector is a shrouded right angle connector with a 2mm pitch. It is supplied by Molex and its part number is 87333-1620.

A suitable connector for mating with this is Molex part number 51110-1651 which is a crimp housing that accepts the crimp part number 20394-8100 – again supplied by Molex.

Connector Pin out

The connector sits against the top surface of the PCB, and has a polarising slot in the top.



All pins on this side of the connector
are connected to ground

ESD protection

All of the Digital I/Os are protected against Electro Static Discharge and over voltage. The devices used are Harris SP723 parts.

This protects the inputs to IEC1004-2 level 4, and provides over voltage limiting to the range 0 to +5V.

McBSP connectors

The McBSP connectors on the HERON2 provide all three of the Multi Channel buffered serial ports of the C6203.

These serial ports can be configured under software control to interface between C6000 processors, or to a variety of other serial port standards.

The HERON2 does not buffer or terminate these signals at all, they are simply connected by PCB track to the connectors. An exception is that McBSP #2 is also tracked to the control FPGA on the HERON2, and is used when the HSB or UMI interrupts are configured using HERON-API. If the HSB or UMI interrupt functions are not used, this McBSP can be used in the same way as the others. When it is used for UMI interrupts it cannot be used as a McBSP.

For details on how to use the McBSPs refer to the TI document "C6000 peripheral reference guide" – from the TI web site http://www.ti.com or from the HUNT ENGINEERING CD.

That same document describes the electrical characteristics of those ports.

Connector type

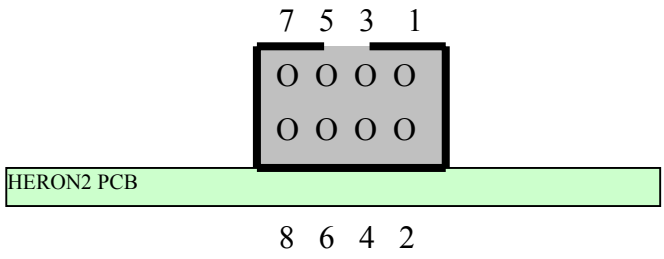
The McBSP connector is a shrouded right angle connector with a 2mm pitch. It is supplied by Molex and its part number is 87333-0820.

A suitable connector for mating with this is Molex part number 51110-0851 which is a crimp housing that accepts the crimp part number 20394-8100 – again supplied by Molex.

Connector Pin out

The connector sits against the top surface of the PCB, and has a polarising slot in the top.

Pin number	Signal name
1	CLKS
2	CLKR
3	CLKX
4	DR
5	DX
6	FSR
7	FSX
8	Ground



ESD protection

The McBSP signals are protected against Electro Static Discharge and over voltage. The devices used are Harris SP723 parts.

This protects the inputs to IEC1004-2 level 4, and provides over voltage limiting to the range 0 to +5V.

FLASH ROM Control Jumper

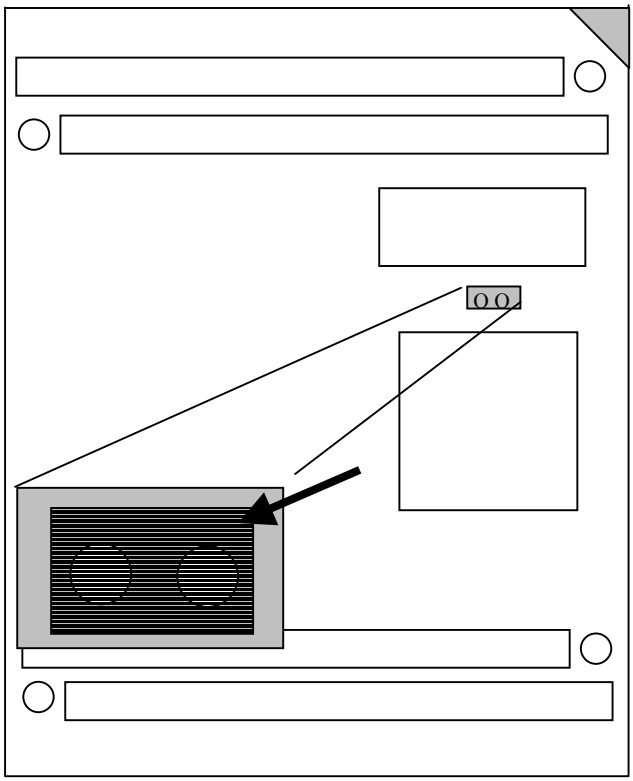
There is a jumper on the HERON2 that controls the operation of the flash memory.

The jumper controls if the FLASH memory is hardware write protected. There is also a separate software write protect mechanism (see earlier section on FLASH ROM) but the Hardware write protect overrides the software protect. The jumper is chosen so that the FLASH ROM is write protected when the jumper is removed, so failure of the jumper means the ROM is write protected.

FLASH ROM write protect

Fitted = writeable

Removed = Write protected



JTAG

The JTAG chain is a serial scan path that needs to be connected through each processor on the system in order to run Code Composer. The design of the HERON module means that the module carrier can determine if a HERON slot does not contain a module, or contains a module that does not implement JTAG (such as a GDIO module).

The carrier card will therefore be automatically configured correctly for the HERON modules fitted.

Uncommitted Module Interconnects

There are some "Uncommitted Interconnect" signals defined by the HERON specification, which are simply connected to all modules.

These are intended to connect control signals between modules, for example a processor module can (via software) drive one of these signals with one of is timer outputs. Then if an I/O module can accept its clock input from one of these signals, it is possible to implement a system with a programmable clock. There will be other uses for these signals that are

module design dependent.

The HERON2 can drive one of these with a Timer output or use one as a timer input, or use one as a digital I/O that can be read and written by the processor, or the processor may receive an interrupt from one. All of these functions are under software control.

See the appendix for details of how to use these connections, but it is strongly recommended that all accesses are made using the HERON-API software. See the later Software section in this manual.

Off-module Peripheral Access

The HERON2 provides the facility to access off-module hardware that may exist on the carrier board into which the module is plugged. This feature may be used, for example, for programming configuration hardware on the carrier board, or for accessing any shared memory that may be available. The Heron2 makes these accesses using the CE1 memory space that is configured as Asynchronous. This allows for programmability of the read and write cycles, to suit the speed of the hardware being accessed. Up to 2Mbytes of off-board hardware may be accessed, as 512k x 32-bit words, from addresses 0x01600000 to 0x017FFFFC.

Heron Serial Bus Controller

The HERON2 has a Heron Serial bus interface that may be used, amongst other things, for network worming and configuration by the Host machine. The 2-wire serial bus connects directly to the module connector pins, from where the carrier board will typically connect the serial bus from all module sites, and the serial bus from the host, together. The interface on the HERON2 may perform master operations as programmed by the HERON2, and it may also respond to slave operations initiated by a different bus master. See Appendix for details of this interface.

The HERON2 has a serial bus so asserts the "Module has serial bus" pin as defined in the HERON specification.

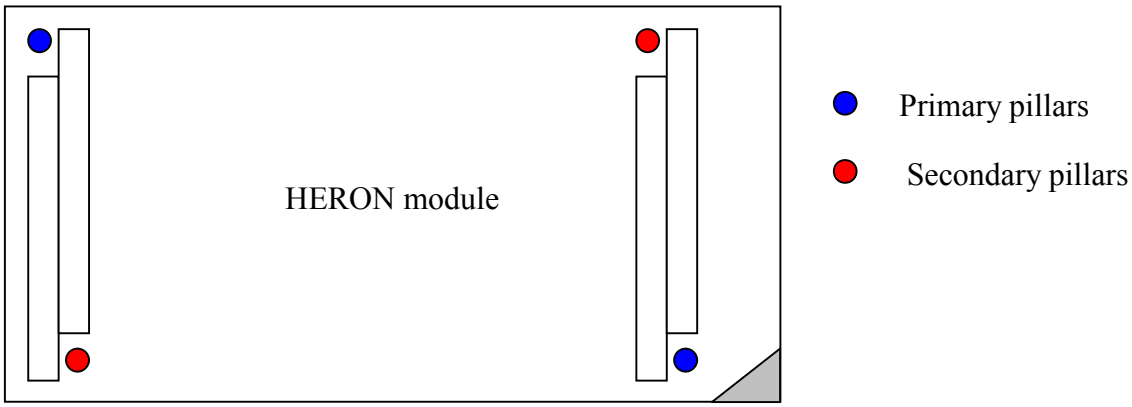
LEDs

Apart from the LEDs on the Digital Outputs, there are four other LEDs on the HERON2. There is one associated with each of the three Power supplies, which must be ON at all times. If they are not a fault has occurred – try powering the system off and back on again. If the Power supply LEDs do not come on contact support.

The final LED is the DONE LED for the control FPGA. This should flash at power on, and then remain off. If it remains on there is a fault with your module.

Fitting HERON modules to your carrier is very simple. Ensure that the module carrier does NOT have power applied when fitting modules, and normal anti-static precautions should be followed at all times.

Each HERON slot has four positions for fixing pillars

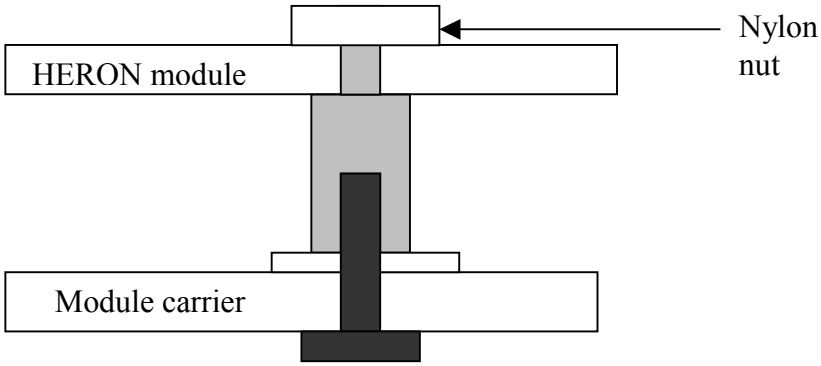


The Carrier card will probably only have spacing pillars fitted to the primary location for each HERON slot. The pillars for the secondary locations will be supplied as an accessory. The reason for this is that the legacy GDIO modules cannot be fitted if the secondary pillars are in place.

The HERON modules are asymmetric about their connectors, so if a module is fitted entirely the wrong way round, the module does not line up with the markings on the carrier card. In particular, notice the triangles on the silk screen of the HERON modules and the HERON slots of the carrier card. These should be overlaid when the module is fitted.

The HERON connectors are polarised, preventing incorrect insertion. So if more than a gentle force is needed to push the module home, check to make sure that it is correctly aligned. Take care not to apply excessive pressure to the centre of the module as this could stress the module's PCB unnecessarily.

Normally the primary fixings will be enough to retain the modules, simply fit the nylon bolts supplied in the accessory kit to the top thread of each mounting pillar.



If the environment demands, the secondary fixing pillars can be fitted to modules that allow their use.

In a HERON system there are many factors that can affect the achievable system throughput. It must be remembered at all times that the part of the system that has the lowest limit on bandwidth will govern the throughput of the system.

The HERON2 can access the HERON carrier's FIFOs in 32-bit mode. It provides a FIFO clock of 75Mhz the processor clock rate.

The HERON2 uses the C6203 Xbus to access the FIFOs which must be accessed using DMA. This leads to the following maximum limits:

Module	Maximum FIFO bandwidth
HERON2 with -C6203 clocked at 300MHz	300Mbytes/second

In fact with the architecture of the C6203 the only way to achieve this bandwidth is to continually transfer data using DMA in the same direction (read or write) and to transfer between the FIFO and internal memory.

Transferring data to/from external memory, or transferring in and out at the same time will cause some idle cycles to be inserted by the processor, so the overall transfer rate will be reduced.

To be in a position to move easily to these future module types it is important that the software that accesses the FIFOs should use the HERON-API software supplied by HUNT ENGINEERING.

In most cases development tools for your system will have been supplied by HUNT ENGINEERING, and will comprise:

HUNT ENGINEERING API	Interface software between the HERON Carrier card and the Host machine software (for all supported operating systems)
HUNT ENGINEERING server/loader	Loader – The tool that will boot a system with an arbitrary number of processors. Server – Tool that provides STDIO access to the Host machine's I/O devices, from within a DSP program.
HUNT ENGINEERING HERON-API	Interface between DSP application software and the HERON FIFOs.
Code Composer Studio	The Integrated Development Environment for developing and debugging the DSP programs. Along with the Compiler, and DSP/BIOS – the real time kernel services
HUNT ENGINEERING plug ins	Helping your development :- Autoconfigure CCS – set up your development environment automatically HERON Project generation – Generate a template project that suits your hardware set up. System Reset – Takes away all the hassles of resetting the system in the correct sequence and ensures that the EMIF is correctly set up after reset. Server/Loader – Integrates the Server/Loader into the development environment allowing you to debug a Server/Loader application simply

DSP program

The DSP program will be written in the C language, possibly with certain functions written in assembler and then called from C. The HERON-API library will be used for communication to other modules, and the host machine.

When the DSP program is written it will be compiled using the TI Code Generation Tools. This compilation can be made simpler by using the Code Composer IDE, which also provides powerful source level debugging features.

The compiled DSP program can be loaded onto the DSPs in two ways:

1. Via Code Composer Studio. This method uses the serial scan chain of the DSP (JTAG) to load the programs onto all of the processors. This can be useful during debug, but cannot be supported in an embedded system, particularly using a host operating system other than windows.
2. Via the HUNT ENGINEERING server/loader. This tool uses a text based configuration file to define which programs should be loaded onto which processors. The loading tales place over the Host FIFOs, after which the STDIO functions of the server can be used. This method is supported under operating systems that are defined as having "target only" support. This means that there is no support for compiling or debugging the DSP code, but it can be loaded via the Server/Loader and HUNT ENGINEERING API.

Code Generation and configuration.

The Code Generation is done using Code Composer Studio. The HUNT ENGINEERING "Create new HERON project" should be used to make a template project that will use the "normal" settings that are good for HERON hardware. This includes setting compiler options, using a template .cdb file (to define the correct memory map) and editing the #include heron2.h line into your source file.

You can then change that project as you wish.

Communications between HERON Modules (HERON-API)

The HERON modules communicate using the bi-directional FIFO connections provided by the HERON Carrier card.

Each type of HERON processor module will have its own method of addressing the FIFOS, so it is important to use the HERON-API software provided with the hardware. This software comes free of charge with your hardware – see the HUNT ENGINEERING CD documentation, the software is installed as part of your HOST API installation and can be found in the heron_api directory under the directory into which you installed your HUNT ENGINEERING API. It can be accessed using the environmental variable HEAPI_DIR, eg %HEAPI_DIR%\heron_api.

The HERON-API follows a similar method to the host side API, bringing a standard set of functions to manage your inter-module communications in the most efficient way supported by your hardware.

The HERON-API determines which HERON module the functions should be compiled to use by which header file you include in the DSP program.

For the HERON2 this line needs to be

```
#include HERON2.h
```

Again an asynchronous model is used allowing a communication to be requested and processing of previous data to continue while the communication takes place. To achieve this, processor DMA is used wherever possible.

HERON-API features that are specific to the HERON2

See the separate documentation on the HERON-API for general details but here are a few points specific to the HERON2 that are important to know.

a) **USE OF WORD BASED FUNCTIONS:-** HERON-API provides the functions HeronReadWord and HeronWriteWord in an attempt to make the input and output of single words more efficient than using a conventional read/write function coupled with a notification such as testio or waitio. The "word" functions are supported by the HERON2, but the use of the Xbus interface to communicate with the FIFOs means that DMA is the only way to access FIFOs. If a dma is free for use then the HeronReadWord or HeronWriteWord functions will still be more efficient for transferring than using a pair of functions like HeronRead and HeronWaitIO. If however there is not a DMA resource available, the Word request will be queued amongst the other transfers, and then the efficiency of the Word functions will be no different from using the normal pair of functions.

b) **USE OF STREAMING I/O:-** Streaming I/O is intended for the access of data streams. It is supported by HERON-API for completeness, although it brings little advantage over the other models (e.g semaphore). The DSP/BIOS part of streaming I/O does not support a mechanism for resetting or restarting the queuing mechanism, so it is not possible to support a closing and re-opening of an SIO HERON-API device. This is not specific to HERON2 and is noted here simply for completeness. It is possible to use SIO and then to use HeronSioClose to free the DMA resources for other uses. It is the re-opening of the SIO device that will fail, either by providing some buffers that were generated by the previous use of the stream, or worse, simply locking up the stream.

c) **INSTALLATION OF NEW ISRs :**- To install a new Interrupt service routine, it can be necessary to use a DMA (if the Interrupt Service Table is in internal program memory). The HERON-API function `HeronInstallIsr` is provided so that your own interrupt vector can be added without needing to claim a DMA from HERON-API specifically for that use. Because the Interrupt vector table can be cached along with other code areas, the cache must be invalidated (flushed) when a new Interrupt vector is installed. The HERON-API function `HeronInstallIsr` takes care of that for you. A side effect of this is that when a new ISR is installed there will be a small performance loss while the cache is re-loaded. The SIO open functions install new ISRs but it is assumed that an SIO open is called only during program initialisation (see section above).

d) **USE OF McBSP#2:-** The HERON2 hardware uses the McBSP #2 to provide interrupts for Heron Serial Bus (HSB) use and for UMI use. This does not allow the McBSP to be used at the same time as these HSB and UMI features. If the UmiInt/SEM/SWI functions or HSB SEM/SWI functions are used then the McBSP #2 cannot, and vice versa if the McBSP#2 is used these functions cannot be used. This does not affect the use of McBSP#0 or 1. Under Code Composer Version 1.2 it is not possible to enable the McBSP #2 as an interrupt source, so these features are not available.

e) **USE OF INTERNAL PROGRAM MEMORY:-** The C6203 has 384Kbytes of internal program memory. The lower 256Kbytes are always mapped memory, whereas the upper 128Kbytes are optionally mapped memory or Cache. The default project settings used when a new project is made using the "Create new project" plug in for Code Composer Studio is to have all code stored in external SDRAM, to use a small

memory model and to have the cache enabled. This is the most general-purpose case. Some projects might be able to fit the entire program into internal memory with the cache off (so there is 384Kbytes of program memory). In this case the .cdb file for a project should be changed to place some of the code into the memory section IPRAM0, and if the cache is set to off then some can be placed in the memory section IPRAM1. To set the cache off you need to go to the Global settings tab of the .cdb file and set Program Cache Control to be "mapped", then this section can be used to place code sections. If a project (with the cache on or off) needs to use *both* internal and external program memory a linker error -- "32-bit PC-relative displacement -12584632 at this location is too large to fit into the 21-bit PC-Relative field" will be issued. This is because there are code calls that are beyond the reach of code compiled using the small memory model. Details of this can be found in the HUNT ENGINEERING application note "Selecting Memory Models and Libraries" where it can be seen that using the –ml1 memory model is required. To achieve that you need to do two things to your project. First set the Project → build options → Advanced → memory models tab to –ml1. Second you need to replace the herons.lib in your project with the heronl1.lib found in the %HEAPI_DIR%\heron_api\lib directory where %HEAPI_DIR% is the directory that you chose when you installed your HUNT ENGINEERING software.

The following sections attempt to cover all likely problems. Please check through this section before contacting technical support.

Hardware

If the Hardware has been installed according to the Instructions there is very little that can be wrong. If the board fails to boot, there can only really be one reason

- It is likely that the "Default Routing Jumpers" are fitted incorrectly or your network file is incorrect. Check the jumper setting and the configuration of the booting software.
- If the module doesn't seem to be functioning, check that the three Power supply LEDs are on, and that the DONE LED for the Control FPGA is off.

Software

As long as the software has been installed using the installation program supplied on the HUNT ENGINEERING CD, there should be little problem with the software installation.

If you have problems then return to one of the example programs supplied with the system.

Server/Loader

If the pre-compiled example programs do not run, then it is likely that there is an error in your network file. If you cannot find one, then try checking the "Default Routing Jumper" settings on your hardware.

If the pre-compiled examples run, but your own program does not, then you should check the following:

- There MUST be a call to Bootloader() at the beginning of your program.
- You MUST link to the Stdio library that has been compiled with the same memory model as your program.

Code Composer

Once Code Composer is correctly installed and working as described in the user manual for your HERON carrier card, there should be little problem at the level of the HERON2.

The most likely problem will be if your program is compiled for a memory map that does not match the HERON2. See the earlier section on Code Generation tools configuration.

HUNT ENGINEERING have performed testing on its products to ensure that it is possible to comply with the European CE marking directives. The HERON4 cannot be CE marked as it is a component in a system, but as long as the following recommendations are followed, a system containing the HERON2 could be CE marked.

The immense flexibility of the HUNT ENGINEERING product range means that individual systems should be marked in accordance with the directives after assembly.

1.The host computer or housing in which the HERON2 is installed is properly assembled with EMC and LVD in mind and ideally should itself carry the CE mark.

2. Any cabling between boards or peripherals is either entirely inside the case of the host computer, or has been assembled and tested in accordance with the directives.

The HERON2 digital I/Os and McBSPs ARE protected against Static discharge, so if the cabling does exit the case, there is suitable protection already fitted.

HUNT ENGINEERING is able to perform system integration in accordance with these directives if you are unsure of how to achieve compliance yourself.

Technical support for HUNT ENGINEERING products should first be obtained from the comprehensive Support section www.hunteng.co.uk/support/support.htm on the HUNT ENGINEERING web site. This includes FAQs, latest product, software and documentation updates etc. Or contact your local supplier - if you are unsure of details please refer to www.hunteng.co.uk for the list of current re-sellers.

HUNT ENGINEERING technical support can be contacted by emailing support@hunteng.co.uk, calling the direct support telephone number +44 (0)1278 760775, or by calling the general number +44 (0)1278 760188 and choosing the technical support option.

Appendix 1 -- Boot Stream Definition

As can be seen by the source code listed on the CD, the HERON2 executes a pre-boot program from FLASH ROM.

The HUNT ENGINEERING Server/Loader tool provides a boot stream in the correct format so most users will never need to consider the boot mechanism. For those who do need to know it is detailed below.

There are multiple sections of the boot stream, all structured identically. A HERON2 module will read and discard complete sections until it receives one that is addressed to it. At that stage it stores it and vectors to it.

Section definition

Word 1	**Module and Carrier ID** – with bits defined exactly as in a later appendix of this manual so that it can be directly compared with the modules Ids register.
Word 2	**Section size** – the number of bytes in this section
Word 3	**Section destination address** – an integer pointer to the location to store this section.
Words 4 to (size>>2)+3)	**Section data.**

This pattern can be repeated as many times as necessary, allowing for multiple sections to be sent to the same DSP.

A zero length section followed by the program entry point, (an integer pointer), terminates the sequence.

Appendix 2 -- Boot prom Program

The source code for the boot ROM program can be found on the HUNT ENGINEERING CD. Choose first “Getting Started” from the CD menu program and then choose “utilities”.

In fact the ROM program supplied can be used as a pre-loader for a user application. This will perform the correct initialisation of the module hardware before continuing to boot a user specified .out file. In this case the .out file can be the same file that is loaded and tested using Code Composer – there are no special requirements to change it as a bootable file, nor to make the addresses for Flash ROM
The program for programming the FLASH ROM can also be found in the same location on the CD along with its source code.

Appendix 3 -- FIFO Registers and Accessing

The HERON FIFOs are addressed at six addresses, where a read from each address reads that Input FIFO, a write to each address writes to that Output FIFO.

It is recommended that direct accesses are not made to these FIFOs, as resource conflicts can occur, and any such code is very specific to the HERON2. The HERON_API software should be used at all times.

The addresses for the FIFOs are as follows:

fifo0	0x4000 0000
fifo1	0x4000 0004
fifo2	0x4000 0008
fifo3	0x4000 000C
fifo4	0x4000 0010
fifo5	0x4000 0014

Each FIFO location is in fact read/write, even though the read FIFO and the Write FIFO may be physically separate devices.

There is also a register where the FIFO flags can be read. The Flags Register may be read at address 0x40 0000.

For the input FIFOs, there are three flags for each channel: The block ready flag, the almost empty flag and the empty flag. They are all active low. I.e. the block ready flag is low whenever a block of data is available to read, the Almost Empty flag is low when the FIFO is almost empty and the empty flag is low when is nothing to read. At reset, the block ready flag will be at "1", the almost empty and empty flags will be at "0".

For the output FIFOs, there are two flags for each channel: The block free flag, and the full flag. These are both active low too. i.e. the block free flag is low whenever there is room for a block of data to be written, and the full flag is low when there is no room in the FIFO at all. At reset, the free ready/flag will be at "0", and the full flag will be at "1".

The bits definition for the Flags Register is as follows:

Bit number	Flag
0	Input FIFO #0 block ready
1	Input FIFO #1 block ready
2	Input FIFO #2 block ready
3	Input FIFO #3 block ready
4	Input FIFO #4 block ready
5	Input FIFO #5 block ready
6	Output FIFO #0 block free
7	Output FIFO #1 block free
8	Output FIFO #2 block free
9	Output FIFO #3 block free
10	Output FIFO #4 block free
11	Output FIFO #5 block free
12	Input FIFO #0 empty
13	Input FIFO #1 empty
14	Input FIFO #2 empty
15	Input FIFO #3 empty
16	Input FIFO #4 empty
17	Input FIFO #5 empty
18	Output FIFO #0 Full
19	Output FIFO #1 Full
20	Output FIFO #2 Full
21	Output FIFO #3 Full
22	Output FIFO #4 Full
23	Output FIFO #5 Full
24	Input FIFO #0 Almost empty
25	Input FIFO #1 Almost empty
26	Input FIFO #2 Almost empty
27	Input FIFO #3 Almost empty
28	Input FIFO #4 Almost empty
29	Input FIFO #5 Almost empty
30	N/U
31	N/U

Appendix 4 – FIFO Interrupts and DMA Synchronisation

It is possible to generate interrupts to the CPU based upon some of the FIFO flags changing state.

The FIFO based interrupts use the external interrupt signals EXTINT4 to 7 to generate CPU interrupts.

The HERON2 allows any of the block ready, block free or almost empty FIFO flags from the Heron interface, to generate a signal on any of the four external interrupt pins of the C6000, INT4, INT5, INT6 and INT7. Multiple sources may be enabled onto an interrupt at any one time, and these are selected via four corresponding External Interrupt Enable Registers, EIER4, EIER5, EIER6 and EIER7. The bits that are set when these registers are written to, determine which interrupt sources are used, in the generation of interrupts, as follows:

Bit number	Flag	Function
0	Input FIFO #0 block ready	Interrupt when there is a block ready to read
1	Input FIFO #1 block ready	Interrupt when there is a block ready to read
2	Input FIFO #2 block ready	Interrupt when there is a block ready to read
3	Input FIFO #3 block ready	Interrupt when there is a block ready to read
4	Input FIFO #4 block ready	Interrupt when there is a block ready to read
5	Input FIFO #5 block ready	Interrupt when there is a block ready to read
6	Output FIFO #0 block free	Interrupt when there is space to write a block
7	Output FIFO #1 block free	Interrupt when there is space to write a block
8	Output FIFO #2 block free	Interrupt when there is space to write a block
9	Output FIFO #3 block free	Interrupt when there is space to write a block
10	Output FIFO #4 block free	Interrupt when there is space to write a block
11	Output FIFO #5 block free	Interrupt when there is space to write a block
12	Input FIFO #0 empty	Interrupt when there is one word to read
13	Input FIFO #1 empty	Interrupt when there is one word to read
14	Input FIFO #2 empty	Interrupt when there is one word to read
15	Input FIFO #3 empty	Interrupt when there is one word to read
16	Input FIFO #4 empty	Interrupt when there is one word to read
17	Input FIFO #5 empty	Interrupt when there is one word to read
18	N/U	
19	N/U	
20	N/U	

21	N/U	
22	N/U	
23	N/U	
24	Input FIFO #0 Almost empty	Interrupt when there is n words to read
25	Input FIFO #1 Almost empty	Interrupt when there is n words to read
26	Input FIFO #2 Almost empty	Interrupt when there is n words to read
27	Input FIFO #3 Almost empty	Interrupt when there is n words to read
28	Input FIFO #4 Almost empty	Interrupt when there is n words to read
29	Input FIFO #5 Almost empty	Interrupt when there is n words to read
30	N/U	
31	N/U	

All bits are cleared at reset.

Setting a bit in this register "enables" that interrupt source.

Register	Address
Interrupt Enable Register INT4	0x40 0004
Interrupt Enable Register INT5	0x40 0008
Interrupt Enable Register INT6	0x40 000C
Interrupt Enable Register INT7	0x40 0010

The C6203 has a mode where when an external interrupt pin is being used to re-trigger a DMA, transitions on the interrupt pin during a DMA burst can be ignored. This method should be used when re-triggering DMAs from block flags.

Appendix 5 -- Uncommitted Module Interconnect

The HERON module has 4 UMI signals defined, which can assume different functions according to the users needs.

The HERON2 can drive each of these pins under software control with its Timer 0 output, the Timer 1 output, logic '0' or logic '1'. For each UMI pin, there are three bits in the UMI_OUTPUT Control Register. There is a UMI Enable bit to select whether the UMI pin is driven by a Timer output or left undriven, and there are two Function bits to select which Timer output or which logic level is used to drive the UMI pin. The functionality of these bits is as follows:

UMI En Bit	Func 1	Func 0	UMI Pin function
0	Don't Care	Don't care	Not driven
1	0	0	Driven by Timer 0
1	0	1	Driven by Timer 1
1	1	0	Driven Low
1	1	1	Driven High

The UMI_OUTPUT Control Register is addressed at 0x40 0020, and has 12 bits as follows:

Bit 0	UMI0 Func0
Bit 1	UMI0 Func1
Bit 2	UMI0 En
Bit 3	UMI1 Func0
Bit 4	UMI1 Func1
Bit 5	UMI1 En
Bit 6	UMI2 Func0
Bit 7	UMI2 Func1
Bit 8	UMI2 En
Bit 9	UMI3 Func0
Bit 10	UMI3 Func1
Bit 11	UMI3 En

All bits in this register are cleared at reset.

When a UMI pin is programmed to be driven, the logic levels are as follows:

SIGNAL	MIN	MAX
Voh	2.4V	
Vol		0.6V

The output current of the device is as follows:

SIGNAL	MIN	MAX
Ioh	-24mA	
Iol		24mA

It is also possible with the HERON2, to connect any UMI pin to the Timer 0 input of the 'C6000, and separately, any UMI pin to the Timer 1 input of the 'C6000.

The 3-bit UMI_TIMER0 Control Register at address 0x40 0024, selects which UMI signal, if any, is used to drive the Timer 0 In of the 'C6000. It is the value written to this register that determines which the functionality of the Timer 0 In pin, as follows:

Register value	Function of Timer 0 Pin.
0	Timer0 IN not driven.
1	Timer0 IN driven by UMI 0
2	Timer0 IN driven by UMI 1
3	Timer0 IN driven by UMI 2
4	Timer0 IN driven by UMI 3

Another register, the 3-bit UMI_TIMER1 Control Register at address 0x40 0028, performs the same function for Timer 1 IN signal of the 'C6000.

Both of these registers are cleared to 0 at reset.

For details of how to use the TINx and TOUTx pins on the C6000, please refer to the TI documentation on the TI web site at http://www.ti.com or on the HUNT ENGINEERING CD.

Finally there is the UMI_INPUT register at address at address 0x40 002C where you can enable UMI based interrupts, and the polarity of them.

Bit0	UMI0 Interrupt Enable
Bit1	UMI1 Interrupt Enable
Bit2	UMI2 Interrupt Enable
Bit3	UMI3 Interrupt Enable
Bit4	UMI0 Polarity
Bit5	UMI1 Polarity
Bit6	UMI2 Polarity
Bit7	UMI3 Polarity

Setting an enable bit to "1" enables the cause as a source of the McBSP#2 receive interrupt. The UMI polarity bits allow either falling or rising edges of the relevant UMI signal to generate interrupts. All bits of this register are zero at reset, which has all interrupt sources disabled, and all UMI polarities are programmed for falling edge interrupts.

The status of the UMI pins can be read from the UMI read register, but it can be possible that a pulse occurs on a UMI to cause an interrupt. In this case there are status bits that are set when the correct type of edge occurs, and are then held for reading by the ISR.

This register is read at the same address (0x40 002C) as the UMI_INPUT register.

Bit 0	UMI0 state
Bit 1	UMI1 state
Bit 2	UMI2 state
Bit 3	UMI3 state
Bit 4	UMI0 edge detected
Bit 5	UMI1 edge detected
Bit 6	UMI2 edge detected
Bit 7	UMI3 edge detected

The edge detected bits are latched, and only set if the relevant UMI has an interrupt enabled on it, and the edge of the type defined by its polarity has happened since the last time you read the register. Multiple edges will be treated as one.

Reading this register clears all of the "edge detected" bits but des not affect the "state" bits.

The OR of the edge detected bits are used to drive the FSR pin of McBSP#2 high. This means if the RINTM bits in the SPCR are set to 10b, with all other configurations set to their default (reset) value, a rising edge on the FSR will cause a processor interrupt on MCBSP2 receive. All that remains it to enable that interrupt onto the processor.

This interrupt cannot be used at the same time as using McBSP2 as a serial port.

The HERON2 module has 4 Digital inputs and 4 digital outputs that can be used by the users program. There are also LEDs connected to buffered versions of the Digital Outputs which illuminate when the Digital Outputs are zero.

The DIGITAL_IO Register is addressed at 0x40 0014, and has 4 read/write bits as follows:

Bit 0	DI/O 0
Bit 1	DI/O 1
Bit 2	DI/O 2
Bit 3	DI/O 3

A write to the register sets the value of the Digital Outputs and a read reads the value of the Digital Inputs.

When a Digital Output pin is driven, the logic levels are as follows:

SIGNAL	MIN	MAX
Voh	2.4V	
Vol		0.6V

The output current of the device is as follows:

SIGNAL	MIN	MAX
Ioh	-24mA	
Iol		24mA

The Inputs have the following characteristics:

SIGNAL	MIN	MAX
Vih	2.0V	
Vil		0.8V

The input current requirement of the device is very low (10uA) but the inputs are pulled high on the HERON2 using a 10K resistor to +5v. This makes the input impedance 10K.

Appendix 7 – ID Register Accessing

There is a register where the HERON2 can read the ID of the carrier and the Module slot ID.

It is recommended that direct accesses are not made to this register, as any such code is very specific to the HERON2.

The register can be read at address 0x40 0018.

The bits in the register are defined as

Bit	Function
0	Module ID bit 0
1	Module ID bit 1
2	Module ID bit 2
3	Module ID bit 3
4	Carrier ID bit 0
5	Carrier ID bit 1
6	Carrier ID bit 2
7	Carrier ID bit 3

Appendix 8 – Heron Serial Bus Register Accessing

There are three locations in the Heron 2 memory map for the Heron Serial Bus (HSB).

Address	Function
0x40 0030	Data. Read/write
0x40 0034	Control register, write Flags register, read
0x40 0038	Target address, write only

The Data location is for reading data bytes that have been received over the HSB from another bus master, and also for writing data bytes that are to be sent. When reading data, the Empty Flag bit in the Flags register may be polled to determine when there is a new data byte to read. Similarly, when writing data, the Full flag in the Flags register may be used to determine when there is room to write another data byte.

Flags Register bit definitions.

Flags Register bit	Function
0	Empty Flag
1	Full Flag
2	Message receiving
3	End of message
4	Message OK

The Empty Flag indicates the status of the incoming data buffer. This bit is active LOW to indicate that the read buffer is empty. The Empty Flag will transition to '1' to indicate that a new data byte has been received.

The Full Flag indicates the status of the outgoing data buffer. This bit is active LOW to indicate that the write buffer is full and cannot yet accept new data. The Full Flag will transition to '1' to indicate when the previous data byte written has been completely shifted out and there is room for the next one. Only when this flag is high, is it safe to write new data.

The "Message Receiving" bit indicates that the slave controller is currently receiving a message.

The "End of Message" bit indicates that the Master controller has sent the last byte of data and generated the STOP condition on the HSB bus, which has been acknowledged, and that the bus is now free.

The "Message OK" bit indicates that the master controller has not detected any error conditions while sending a message, for example losing the bus to another bus master.

Control Register bit definitions.

Control Register bit	Function
0	N/U
1	N/U
2	N/U
3	N/U
4	N/U
5	"Send Message" Bit
6	"Message receiving" interrupt enable

The "Send Message" bit is used to generate the HSB Start and Stop conditions at the head and tail of a message respectively, and also to send the address of the slave controller that the message is destined for. The address used is the value set in the target address register. These operations are all transparent to the user, all that is required is to set the "Send Message" bit before sending the first byte of data, and then to clear the "Send Message" bit after the last byte has been sent.

The interrupt enable on the "Message receiving" enables the processor to be interrupted when the "Message receiving" bit is first set for a message.

These bits are cleared at reset.

When enabled the Message receiving bit is used to drive the FSX pin of McBSP#2 high. This means if the XINTM bits in the SPCR are set to 10b, with all other configurations set to their default (reset) value, a rising edge on the FSX will cause a processor interrupt on MCBSP2 transmit. All that remains it to enable that interrupt onto the processor.

This interrupt cannot be used at the same time as using McBSP as a serial port.

Module Type Query

The HERON2 hardware will respond to the "Module type query" HSB message without interaction from the processor. In this case the "Message receiving" bit will not be set.

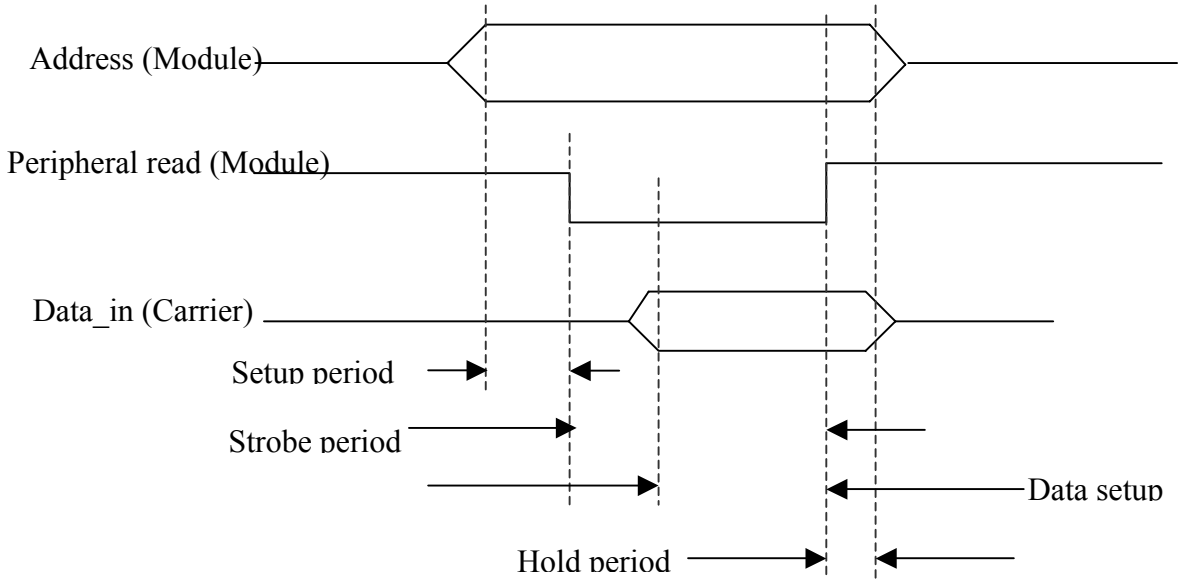
Appendix 9 – Off-module Peripheral Access

The Heron2 provides the facility to access off-module hardware that may exist on the carrier board into which the module is plugged. This feature may be used, for example, for programming configuration hardware on the carrier board, or for accessing any shared memory that may be available.

The Heron2 makes these accesses using the CE1 memory space that is configured as Asynchronous. This allows for programmability of the read and write cycles, to suit the speed of the hardware being accessed. The user can address the off module peripherals at address 0x160 0000 → 17f ffff.

All three periods of the cycle, setup period, strobe period and hold period, are programmable in steps of 1 CPU clock period, and are separate for reads and writes.

Peripheral Read Access



Setup period

The setup period is defined as the period between address generation and the assertion of the peripheral read (or write) signal. It is controlled by a 4-bit field in the CE1 Space Control Register, and can therefore be up to 15 CPU cycles in duration. (The minimum is 1, and the first access of a burst will always be 2). The address is routed through the control FPGA on the HERON2, which can have a 6ns propagation delay. The Control signals are generated in that same control FPGA and can have a 9ns propagation delay.

This means that the combination of the C6203 timings and the hardware of the HERON2 module means that the minimum setup period will be ((n*3.3ns) - 11ns). Thus the minimum setting that should be used is 4.

Strobe period

The strobe period is defined as the duration of the peripheral read (or write) signal. It is

controlled by a 6-bit field in the CE1 Space Control Register, and can therefore be up to 63 CPU cycles in duration. (The minimum is 1). The delay through the Control FPGA should be similar for the assertion and de-assertion of this signal so can be programmed as n*3.3ns. The HERON2 module has circuitry that use the ardy signal to extend accesses to the flash memory, if the other accesses in the same CE space (which is the off module peripherals) use a Bus Control Register that sets the cycles too short for the FLASH. This circuitry only works if the Strobe setting is more than 2cycles.

Hold period

The hold period is defined as the period between the peripheral read (or write) being de-asserted, and the address becoming invalid. It is controlled by a 2-bit field in the CE1 Space Control Register, and can therefore be up to 3 CPU cycles in duration. The Address is delayed through the control FPGA by up to 6ns, and the control signals by up to 9ns, Thus the combination of the C6203 timings and the HERON2 hardware means that the result of programming this register to n is that the minimum hold period is ((n*3.3ns)-9ns) so the register should never be set to less than 3 cycles.

Data set-up

On a write cycle the data is driven by the HERON2 with the same timing as the address.

The data is delayed by up to 6 ns through the control FPGA, so on a read, the combination of this and the C6302 means that the HERON2 requires that the data set up time to the end of the strobe period be 7ns.

CE space control register

The register used to program these settings is internal to the C6203. Refer to the C6000 peripherals guide for details. Using the minimum values stated above the BCR value would be 0x40b40223.

Example.

A carrier board, that the Heron2 is plugged into, has a bank of 15nS asynchronous, static RAM. The Peripheral Read is connected directly to the RAM's output enable pins, and the Peripheral Write is connected directly to the RAM's write enable pins. The RAM's Chip Enable is assumed to be permanently active.

Read cycle.

Start at t = 0. C6203 cycle starts.

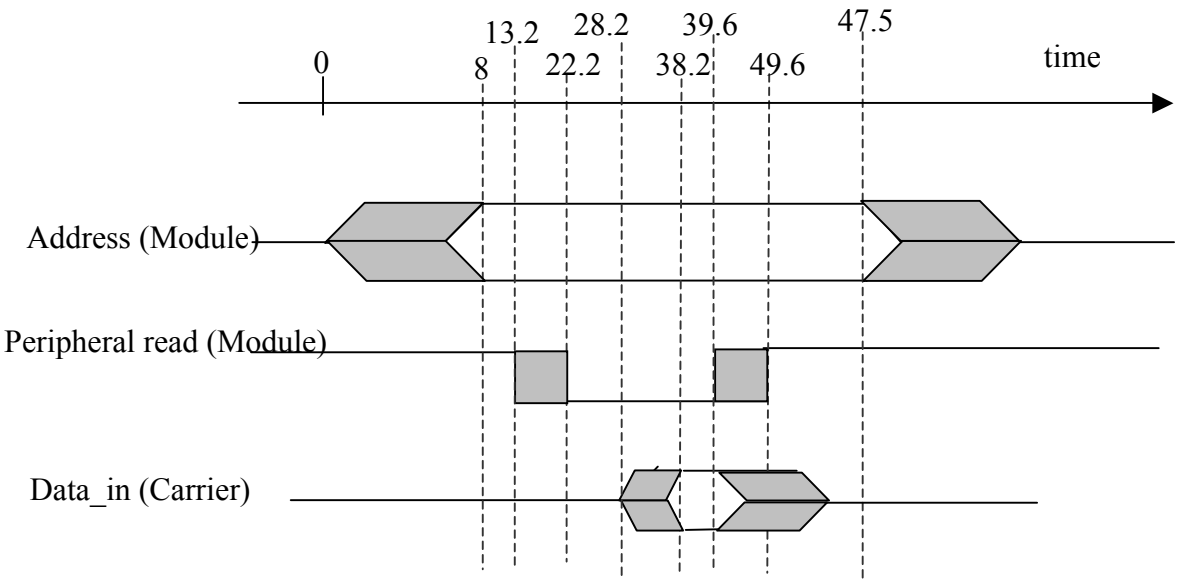
Peripheral address can be driven after 0ns, but is guaranteed after 8ns.

If we have Setup period programmed to 4 cycles, the peripheral read can fall at t=13.2, but is guaranteed to be asserted at t=22.2

The access time of the SRAM is 15ns, and the data setup must be 7ns, so we need the strobe low for at least 22ns. This means we must program the strobe period to be 8 cycles. This means that the data can be valid at t=28.2, but is guaranteed to be valid at t=38.2.

Although the Peripheral Read signal can rise as early as t=39.6, it will only do so if the data was valid at t=28.2 (because the Peripheral read fell at t=13.2).

If the hold time is set to 3 cycles, the address will remain valid at least until t=47.5 when the next cycle could start



Example read cycle with setup =4 cycles, strobe =8 cycles and hold =3 cycles

Write cycle.

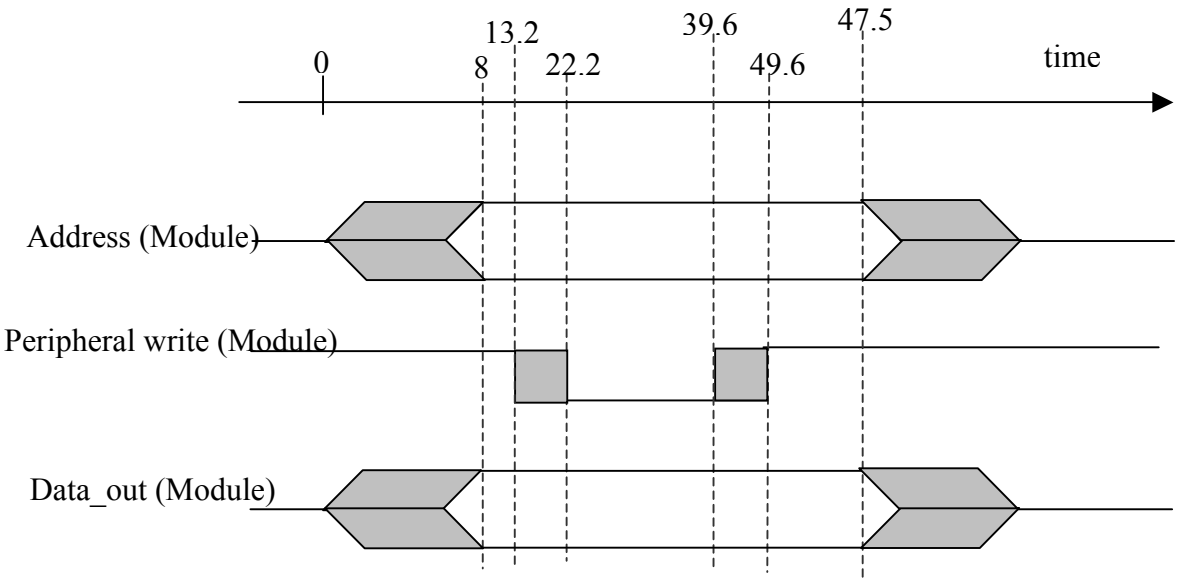
Start at t = 0. C6203 cycle starts.

Peripheral address and the write data can be driven after 0ns, but is guaranteed after 8ns.

If we have Setup period programmed to 4 cycles, the peripheral write can fall at t=13.2, but is guaranteed to be asserted at t=22.2

With the Strobe time set to 8 cycles, the Peripheral Write signal can rise as early as t=39.6, it will only do so if it fell at t=13.2.

If the hold time is set to 3 cycles, the address and data will remain valid at least until t=47.5 when the next cycle could start



Example Write cycle with setup =4 cycles, strobe =8 cycles and hold =3 cycles