



CBC2 (CMS Binary Chip 2) User Guide 1.1

Author:

Davide Braga
email: davide.braga@stfc.ac.uk
tel: +44(0)1235 778243

Date:

V1.1 21/03/2013
V1.0 23/07/2012



1. *Introduction*

The CBC2 is designed for the readout of the silicon microstrip tracker of the CMS experiment on the HL-LHC. The HL-LHC will accelerate two beams of charged particles in opposite directions around a circular particle accelerator based at CERN in Geneva. The particles are concentrated into bunches and the paths of these bunches cross at intervals around the ring. The CMS experiment is based at one such point and the tracker forms the inner region of the experiment – its purpose is to track charged particles produced in the bunch collisions. The inner region of the tracker comprises pixellated silicon detectors whilst the outer region comprises silicon strip detectors.

The CBC2 chip reads out the charge generated by ionising events within the silicon strips of the CMS detector. It converts these events into a “hit” or “no hit” binary value for each of the channels. These ionising events are synchronised with the bunch crossing event interval of 25ns and without knowing which events are suitable for readout the chip must store the data from each event up to a maximum of 256 bunch crossing intervals (6.4 μ s for a 25ns clock). This time is known as the trigger latency, and is the time it takes the external system to decide which of the events are worth reading out.

When the CBC2 receives a trigger signal the event stored one trigger latency previously is read out into a 32-event-deep data buffer where it is stored until it can be read out from the chip. The data buffer is required because data is read out serially from the chip at no faster than the clock. For a clock rate of 40MHz, reading 254 channels, a time stamp, plus a few other bits requires 6.65 μ s. At the same time the CBC2 must continue to take data every bunch crossing.

The average trigger rate is 100kHz (10 μ s). However, due to the random nature of the trigger, the data buffer must be deep enough to cope with several triggers arriving faster than 100kHz. Provided the data from one event is read out quicker than 10 μ s then the data buffer only rarely fill up and events be lost.

Since there will be several thousand CBC2s operating within the small volume of the tracker, the power consumption must be minimised as far as possible. During operation of the HL-LHC the environment will be highly radiated with charged particles requiring the electronics to be designed using radiation tolerant techniques, both for total dose and for Single Event Transients (SETs). The chip is fabricated in 130 nm CMOS.



Version Control:

V1.1:

- Table 36 comment
- Table 25 -- Icomp: Page 1 Address 00001001Table 25: Icomp default values
- Table 2: not ± 10 but ± 8 strips
- Figure 25. Data Timing: 5 clock cycles not 7



2. *Specifications*

2.1 *Front End*

- 254 channels + 2 dummy channels (NB: dummy outputs not available).
- C4 bump bonding, 5on10 (250 μ m pitch), plus one columns of wire-bondable pads for testing.
- Reads both electrons and holes (selectable)
- Charge collection time: less than 10ns.
- Amplifier peaking time: less than 20ns.
- Amplifier noise: less than 1000 electrons up to 5pF detector capacitance and leakage current of 1 μ A.
- Detector leakage current noise: less than 500 electrons up to 1 μ A leakage
- Time walk: (difference in timing in the output of the comparator at 0.5 VDD for 1.25fC and 10fC signals and a comparator threshold of 1fC) \leq 16ns (from the ABCD3T specification).
- 8 bit trim of postamp offset for comparator threshold matching.
- Overload recovery: response to normal signal $<2.5\ \mu$ s after a heavy ionising particle (HIP) of 4pC.
- Coincidence logic for finding hits associated with “hard” (high-Pt) tracks:
 - Signal path parallel to and independent from full-readout data;
 - Individual channel mask for noisy channels;
 - Clustering of adjacent hits from the same sensor and suppression if wider than a programmable width (up to ± 2 strips);
 - Stubs finding within a programmable window of up to ± 8 adjacent strips;
 - Correction of the geometrical offset due to the position of the detector in the R- φ plane;
 - Slow/serial readout of stubs candidates.

2.2 *Pipeline, Buffering and Logic*

- Pipeline depth (latency) from 1 to 256 clock cycles (6.4 μ s at 40MHz).
- Latency and FIFO overflow error checking.
- Capable of buffering 32 triggers awaiting readout.
- Critical control blocks designed with SEU tolerance.
- Registers with triple voting logic and external refresh to avoid accumulation of errors.

2.3 *Control*

- Slow control via an I²C interface used to configure the chip.
- 4 dedicated lines for fast control: T1_TRIGGER, FAST_RESET, TEST_PULSE, I2C_REFRESH.

2.4 *Input/Output*

- Fast Input: SLVS (Scalable Low Voltage Signalling)
- Fast Output: SLVS
- Slow control: I²C pads.



2.5 *Power*

- Analogue supply (VDDA) 1.1 to 1.2V.
- An on-chip DC-DC converter can be used to generate 1.2V from 2.5V.
- An on-chip LDO regulator can be used to supply VDDA at 1.1V
- DC-DC and LDO selectable using bond-pad configuration.
- Digital supply (VDDD) up to 1.2V.
- Power supply rejection as good as possible for a single ended input stage.

3. *Functional Description*

3.1 *Summary*

The CMS Binary Chip 2 (CBC2) is a full-scale prototype for the front end readout ASIC for silicon strips in the CMS tracker for HL-LHC. The original LHC readout chip (APV25-S1) had analogue, non-sparsified readout architecture. For HL-LHC, a binary non-sparsified readout has been chosen as the target architecture for the CBC2.

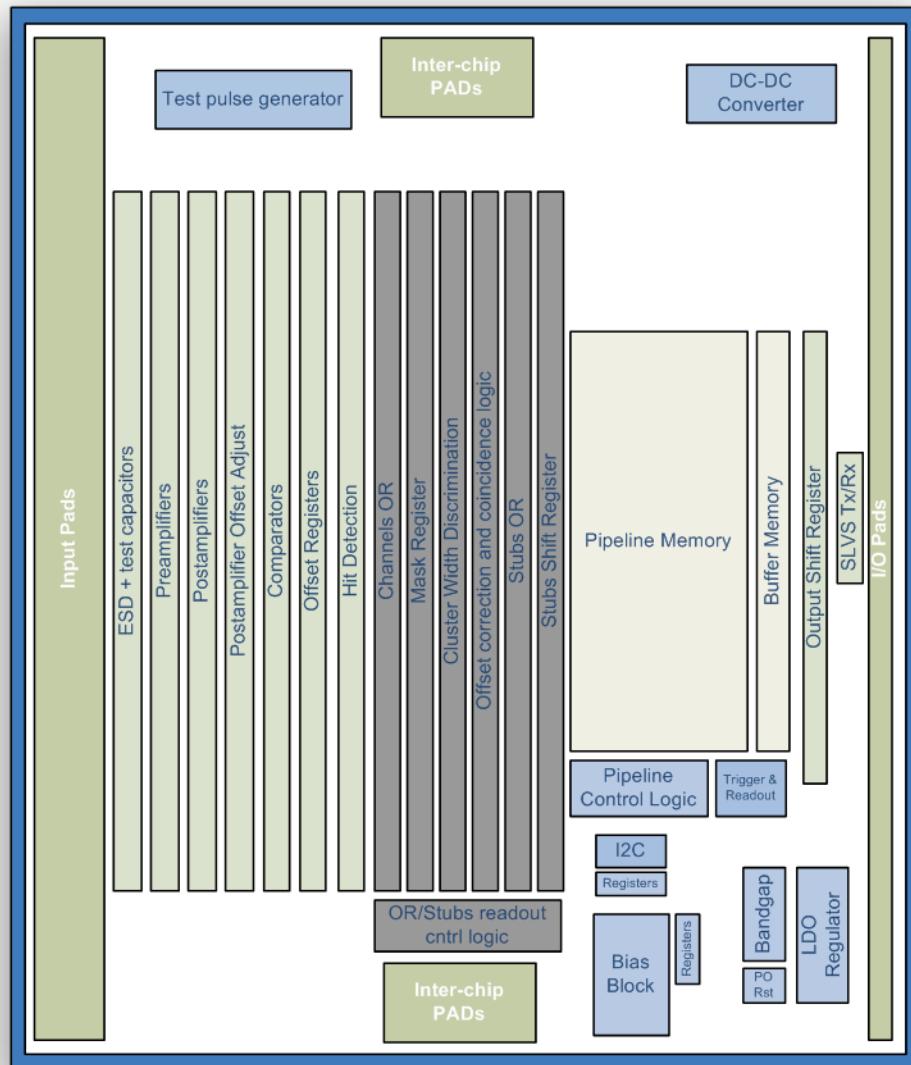


Figure 1. CBC2 block diagram

Non-sparsified data simplifies the readout architecture and has the added benefit that it can be emulated off-detector making it easy to spot chips which become out of synchronisation with the rest of the system. A binary architecture has been chosen to reduce the amount of data that needs to be processed.

The ASIC is designed to instrument double-layer modules, with the input channels alternatively connected to lower and upper detector. But CBC2 could also be used to instrument a single detector, with all the 254 inputs connected to the same detector, in which case the coincidence logic can be disabled.



The design has been partitioned to perform the requirements as shown in Figure 1. The front end consists of 254 channels of charge preamplifiers followed by gain amplifiers. Comparators detect any signals over threshold and pulse detection logic generates “hit” signals. Binary data from the front-end comparators are synchronized with the bunch crossing clock and then follow two parallel paths: one for trigger readout (red in Figure 2) and the other for stub finding (green). In the former data are continuously written into a 256-deep FIFO pipeline RAM, for a maximum trigger latency of 6.4 μ s. In the other data path several blocks of combinatorial logic identify stub candidates before writing them into a shift register for serial read out.

The *Pipeline Control Logic* (PCL) sequences the writing of data from the *Hit Detection* circuit into the *Pipeline Memory*. The PCL controls the transfer of data from the *Pipeline Memory* into the *Buffer Memory* when an external trigger signal is received.

Data is held in the *Buffer Memory* until it can be read out. This will depend on how many previous triggered data are also held in the buffer. When the data is read out, it is serialised together with the column address (time slice) from which it originally came and some information bits. This function is controlled by the *Trigger & Readout logic* and data read out through the *Output Shift Register* and *SLVS* output.

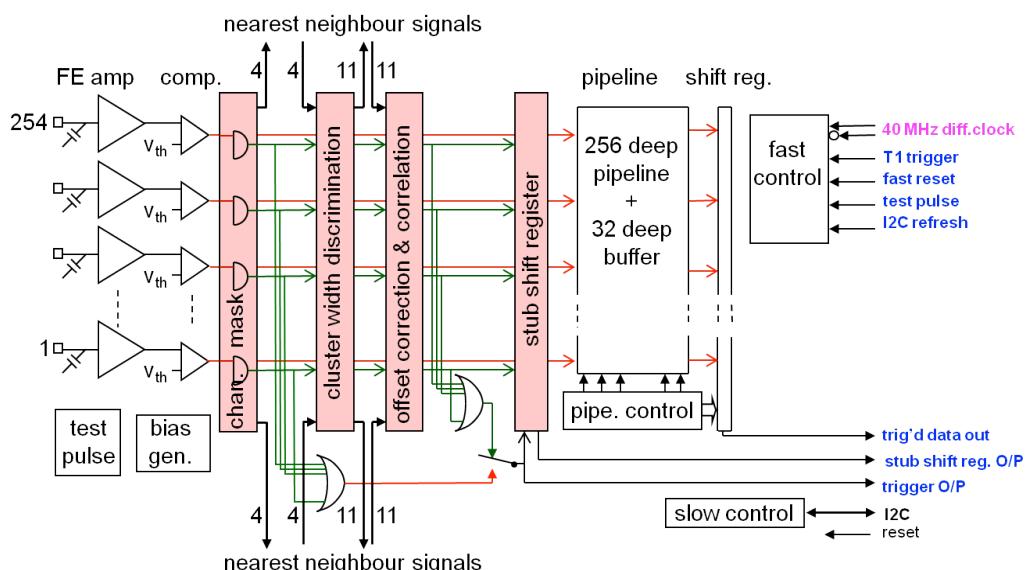


Figure 2. CBC2 signal path.

A Programmable *Bias Block* is used to supply bias voltages and currents for the front-end analogue electronics. A programmable *Postamplifier Offset Adjust* allows 8-bit control for comparator offset matching. A further *Register* configures the operation of the chip. These are all programmed via an *I2C* interface.

A band gap provides a reference voltage for the bias circuit, and this is initialised by a power-on-reset circuit.



A *Low Drop Out* voltage regulator (LDO) can be used to supply the analogue front end and bias generator with 1.1V from the 1.2V supply. This increases power supply rejection, especially if the DC-DC Converter is used to supply the 1.2V from 2.5V.

3.2 Front End

The CBC2 front end is shown in Figure 3. The input is bonded to one strip of the detector. Charge generated by an ionising event in the strip is read out by a *preamplifier* and integrated onto a feedback capacitor. The feedback capacitor is discharged by a resistive feedback network which is selectable to optimise the circuit for both electrons and holes.

The resulting voltage pulse from the preamplifier is further amplified by a capacitive gain *postamp*. A large value feedback resistance stabilises the amplifier. This feedback network is also selectable depending on the polarity of the input charge. To compensate for any mismatch in amplifier and comparator thresholds the *postamp* has a programmable offset controlled using a differential current Ipaos. These currents are programmed using an 8 bit register in each channel

The comparator detects signals over a defined threshold and will produce a digital “1” pulse during this period. A polarity select circuit is used to keep the polarity the same no matter whether electrons or holes are being read out. A 4 bit programmable level of hysteresis is also available.

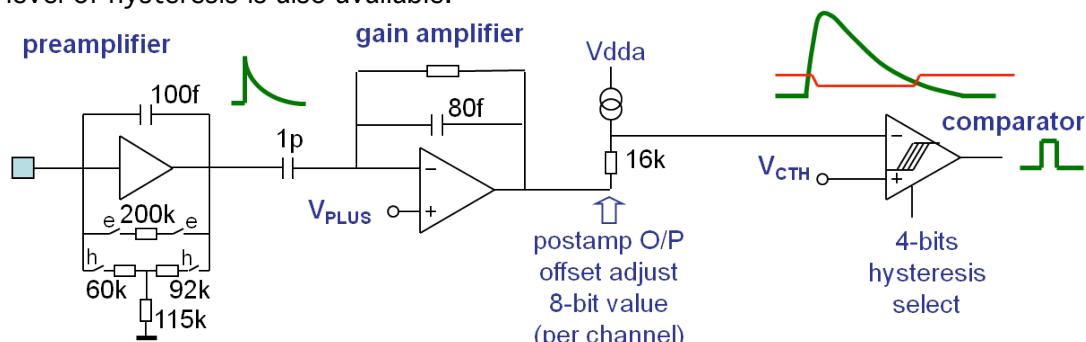


Figure 3. CBC2 Analogue Front End

3.3 Preamplifier

The *preamplifier* (Figure 4) consists of a single ended cascode amplifier, a source follower at the output, and a feedback capacitor and resistor network.

The open loop voltage gain (A_v) must be sufficiently high such that most of the charge generated by an ionising event is removed from the detector. The detector capacitance (5p) and any parasitic capacitance at the input, must be small compared to $A_v \cdot C_f$ so that most of the charge is integrated onto the feedback capacitor C_f . In



addition, the feedback capacitor connects to the input of the source follower and therefore helps in compensating the amplifier.

The integrated charge is discharged through the feedback network. If electrons are being read out the switches “e” will be closed and a single resistor of 200k forms the feedback resistance. If holes are being read out then switches “h” will be closed and a t-network forms the resistive feedback.

The resistance of the feedback network must be sufficiently large such that the time constant RfC_f is not so small compared to the rise time of the signal such that signal is lost. However, the feedback resistor must not be so large that any offset due to detector leakage current reduces the dynamic range of the amplifier.

The resulting voltage pulse is driven to the gain amplifier with the low output impedance of the source follower which also provides a voltage step which ensures the cascode transistor remains in saturation. The cascode bias V_{pc} must be supplied.

The single ended nature of the amplifier does make it susceptible to noise on the power and ground lines.

The design makes use of the triple well process allowing the NFET substrates to be biased independently from ground.

The nominal current used by the preamplifier is $90+10+25 = 125\mu A$.

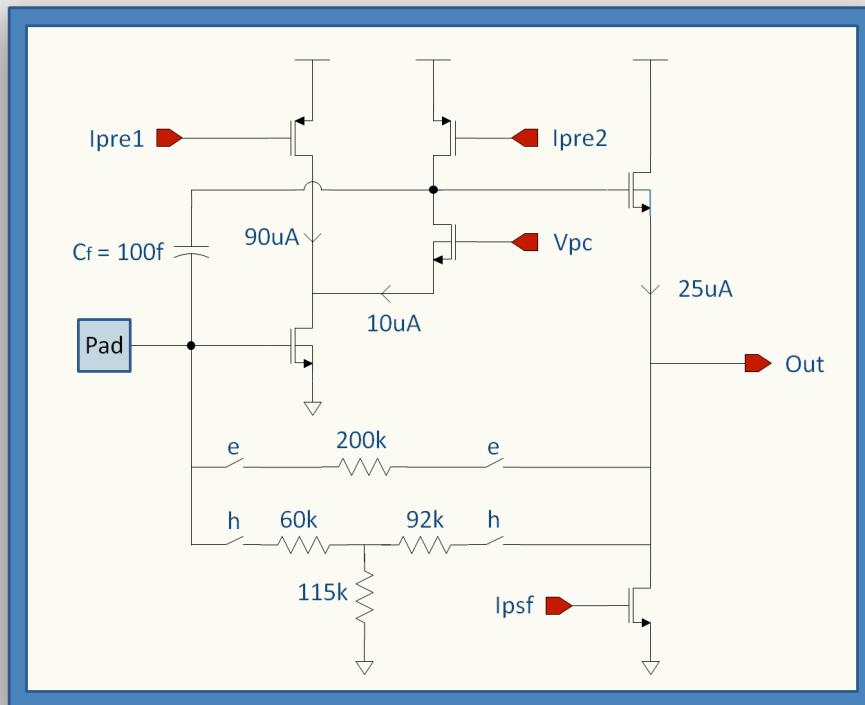


Figure 4. Preamplifier Schematic

3.4 Poststamp

The *poststamp* (Figure 5) is comprised of a differential amplifier – an input capacitor C_{in} and feedback capacitor C_f provide the voltage gain. DC stability is provided by a feedback network, selectable depending on the polarity of the signal. Two diode connected PFETs are used in the first stage of the amplifier to limit the differential swing to speed recovery time for very large (HIP) signals. A capacitor C_c compensates the amplifier, providing stability.

A resistor in the output branch allows the offsetting of the poststamp output voltage. This is done by adjusting the differential currents I_{paos1} and I_{paos2} . These are set by loading an 8 bit control register in the channel.

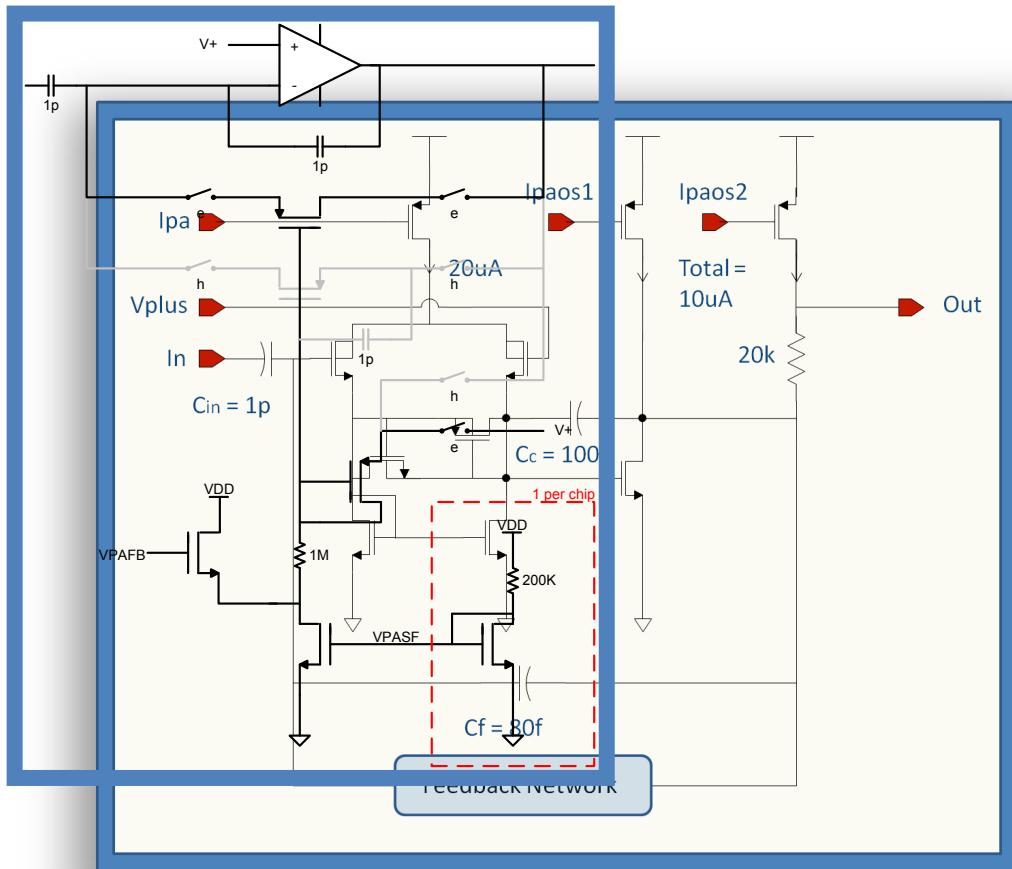


Figure 5. Postamplifier Schematic

The nominal current in the postamplifier is $20 + 12.7 = 32.7\mu A$

The feedback network in both case of polarity comprises a long PFET biased with a small current. However the connections are slightly different for electrons and holes.

Figure 6 shows the configuration for electrons. The 3 switches "e" connect a current mirror in the configuration shown. Since the output signals are negative going, the sources of the current mirror transistors must be connected to the higher potential (V_{plus}). A further circuit in the bias generator (common to all channels) can be used to adjust the current in the feedback circuit.



Figure 7 shows the configuration for reading out holes. In this case the output signal is positive going so the sources of the current mirror transistors must connect to the output. A 1pF capacitor is used to ensure that the gates of the transistors in the current mirror follow the output to maintain linearity.

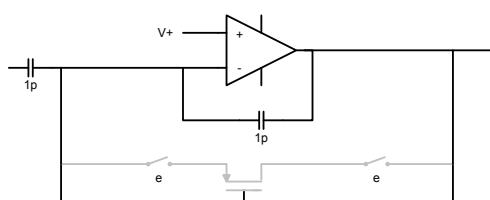


Figure 7. Postamp "holes" configuration

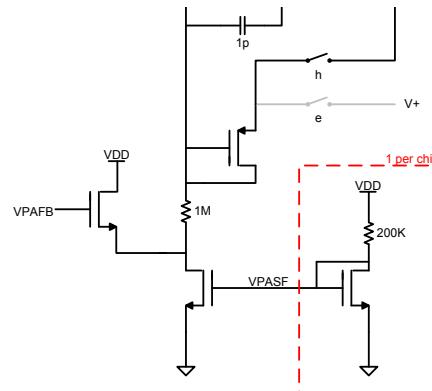


Figure 6. Postamp "holes" configuration



3.5 Comparator

The *comparator* comprises a simple 2-stage differential architecture as shown in Figure 8 followed by some logic. The differential stage is powered from the analogue domain, and the logic from the digital domain. The polarity of the output is selectable depending on whether electrons or holes are being read out – the output from the comparator should always be positive for a signal.

The comparator will have a different response time depending on the size of the input signal. This time walk is mostly dependent on the rise time of the input signal and the gain of the comparator. The time walk should be less than 16ns between a 1.25fC and a 10fC signal when the comparator threshold is set at 1fC.

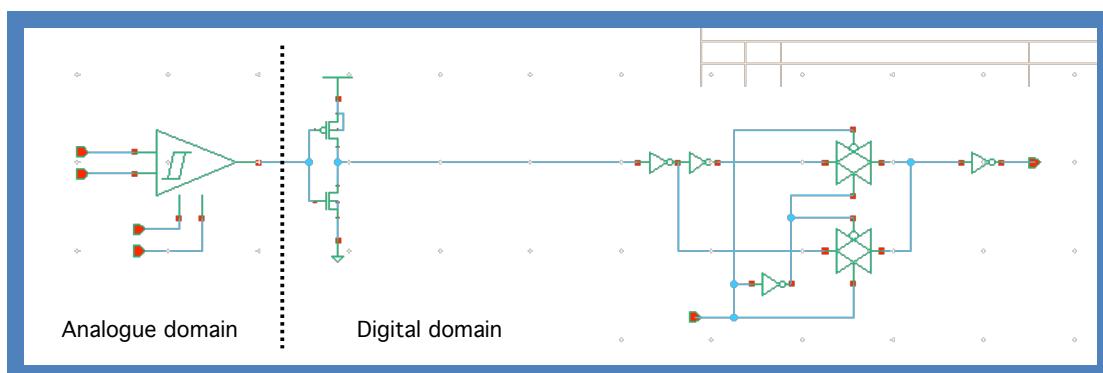


Figure 8. Comparator

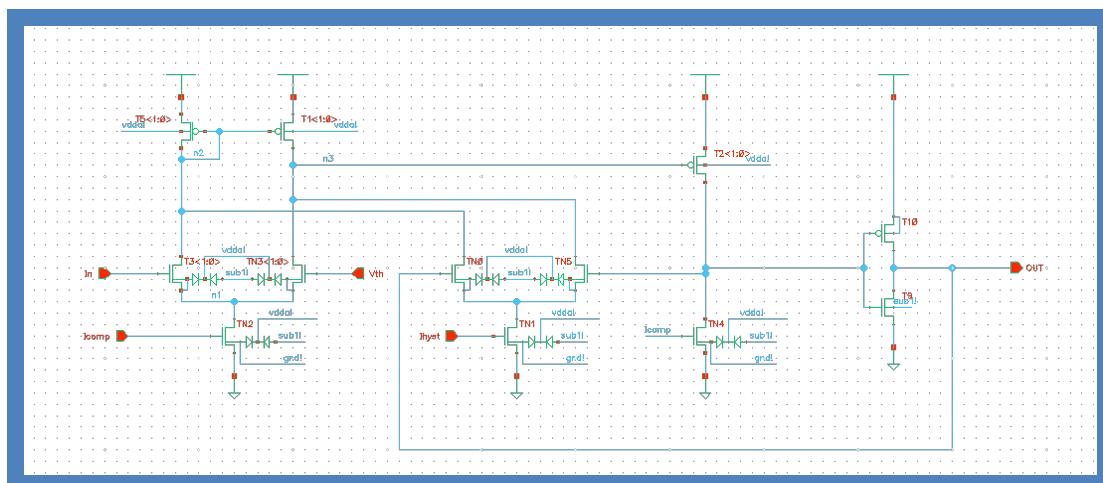


Figure 9. Comparator OPAMP with internal hysteresis

The comparator also has programmable hysteresis. This is achieved by an additional differential couple biased with lower current. The amount of hysteresis is controlled



globally by programming a 4 bit register. The default setting is for maximum hysteresis.

3.6 Postamp Offset Adjust

The comparator threshold to each channel is set globally. In order to compensate for threshold mismatches in the comparator input transistors producing different input offsets from channel to channel it was decided to adjust the level of the input signal to the comparator rather than the comparator threshold voltage V_{Cth} . A programmable, differential current is used to bias the output branches of the postamp. The voltage at the output of postamp is offset by adjusting the current through the 20k resistor. Using the default setting for I_{paos} , this gives a range of 0 to +200mV of offset (Figure 10). The default offset is 100mV. The nominal current is 14uA.

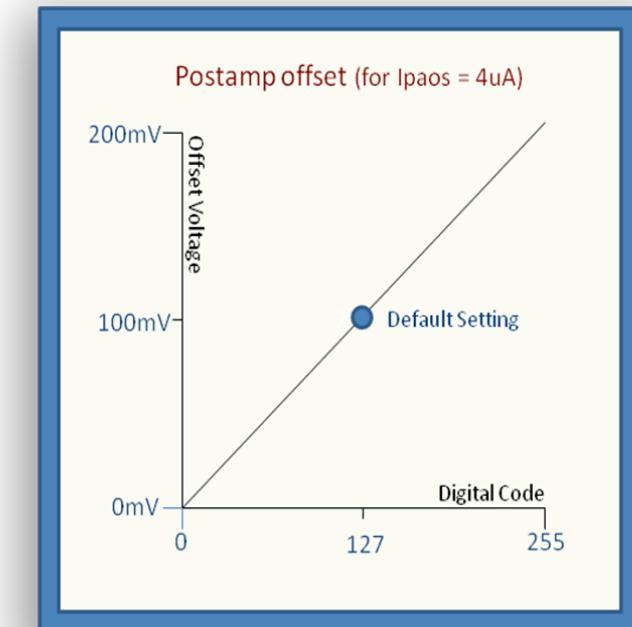


Figure 10. Offset Adjust Range

3.7 Hit Detection Logic

Some logic is required after the comparator to synchronise its output to the clock – the *Hit Detection Logic*. Two modes of operation can be selected. The first mode just passes a synchronised version of the comparator output through to the Pipeline RAM. The second mode detects the rising edge of the comparator and produces a pulse of one clock cycle's length no matter what the length of the comparator signal. In addition, the *Hit Detection* can be completely switched off using a control signal.

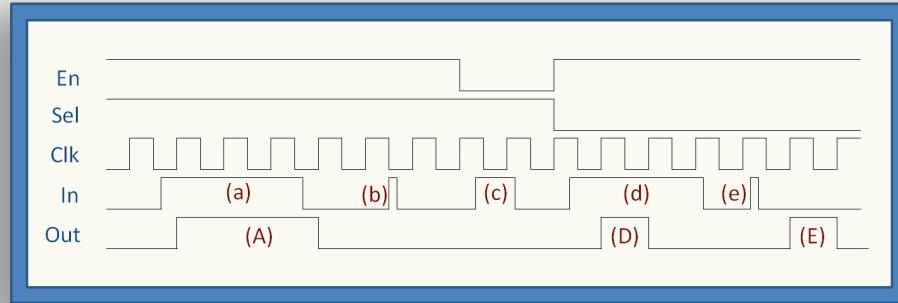
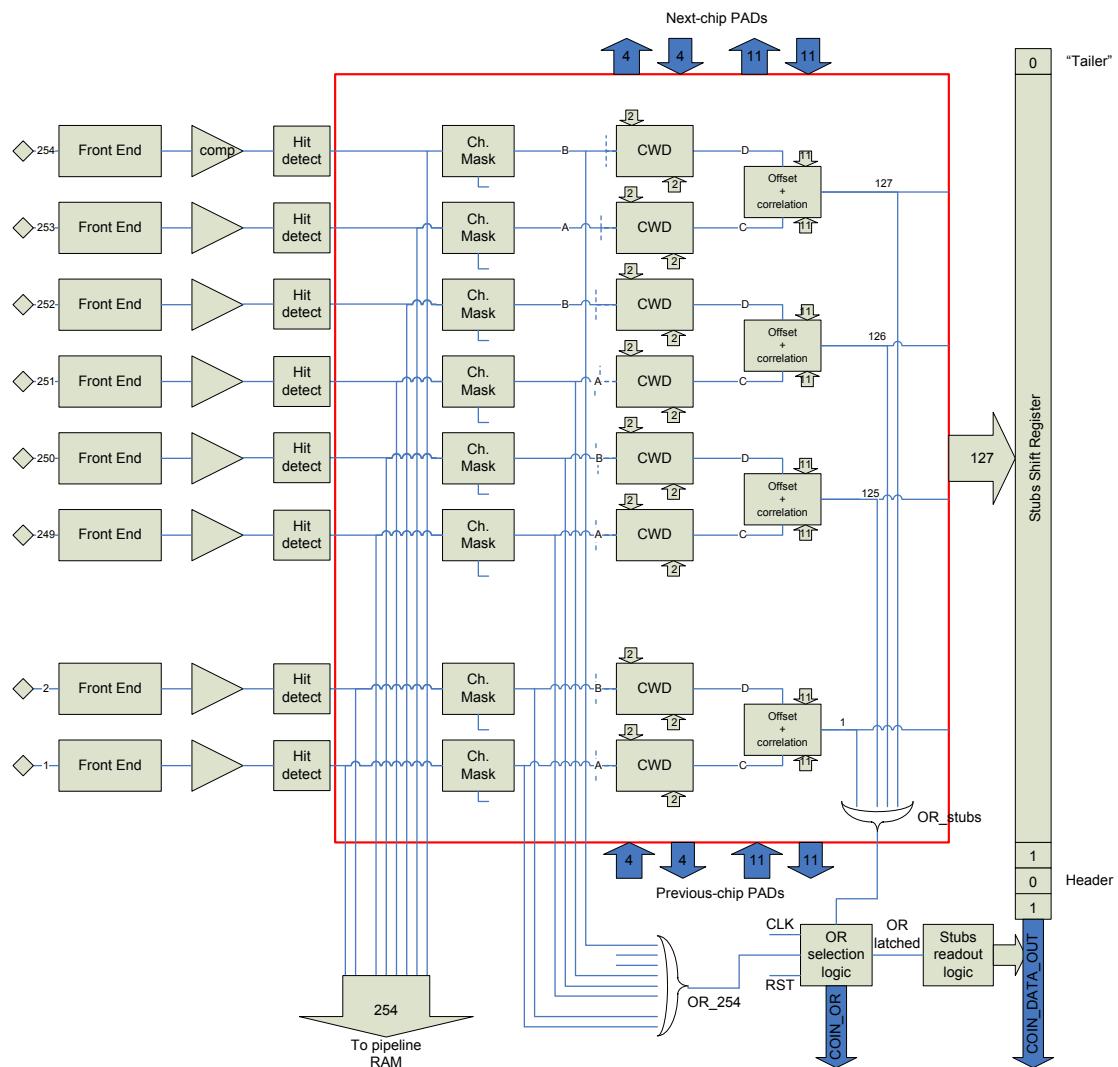


Figure 11. Hit Detection Logic

Figure 11 shows the output from the *hit detection logic* for several cases of input. For





case (a) and (b) the circuit is enabled ($EN=1$) and $SEL=1$ which means a synchronised version of the input should pass through to the output. This is true of (a), however (b) was not high for enough time to be synchronised to the clock. In case (c) the circuit is disabled ($EN=0$) and nothing passes to the output. In cases (d) and (e), the circuit is enabled and $SEL=0$. This means that all comparator pulses are increased or decreased in length to be one clock cycle in length. The output from the Hit detection circuit feeds directly into the pipeline RAM.

3.8 Coincidence/stub-finding Logic

The operation of the stub-finding logic is based on a simple procedure: the first stage rejects wide clusters of hits on both inner and outer sensors; subsequently for every valid cluster on the inner sensor the logic looks for a hit in a coincidence window on the outer sensor. If a hit is present within this window, the inner strip is considered a valid stub.

The binary outputs of all the coincidence logic channels is ORed, and, when a stub is found, they are then latched into a shift register and read out at 40MHz as described in () .

The coincidence logic is structured into combinatorial blocks, which repeat every two adjacent channels (corresponding to one inner strip and the outer strips directly above it).

Noisy channels can be individually masked from the coincidence logic, without affecting the values written into the pipeline RAM.

3.8.1 Cluster-width discrimination (CWD)

The first stage of the combinatorial logic analyses adjacent strips from the same sensor to reject wide clusters of hits, which are typically associated with low-momentum tracks. 2 bits in a global register (address 00011000) set the value of the maximum cluster width to one, two or three adjacent strips. Clusters wider than this value are suppressed. The same register allows for the CWD to be “bypassed” and to pass every single hit to the subsequent stage.

Table 1: CWD settings (register 00011000)

Register value <b7...b0>	CWD width
--------------------------	-----------

Figure 13. a) Φ -shift offset; b) effect of offset on coincidence window

00xxxxxx	output=input (CWD bypassed) default state
01xxxxxx	reject clusters wider than 1 strip



10xxxxxx	reject clusters wider than 2 strips
11xxxxxx	reject clusters wider than 3 strips



At the output of the CWD only the central strip of a cluster is active; in the case of a two-strip cluster, the centre is assigned to the strip with lower address.

3.8.2 Φ -shift correction and correlation logic

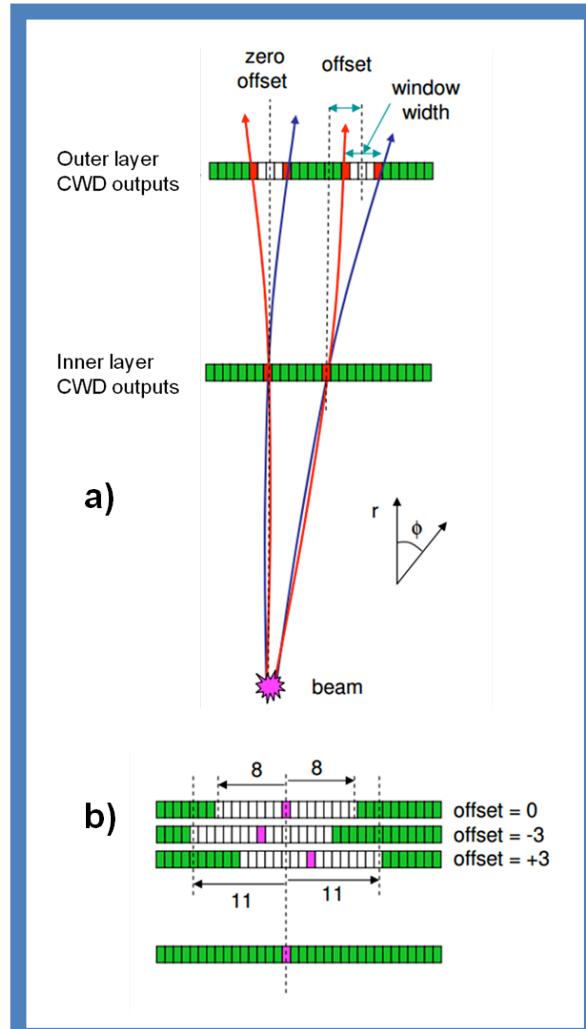
For every central strip of a valid cluster in the inner layer, the logic block following the CWD looks for valid clusters in a coincidence window in the outer layer. If there is any, then the output is true which indicates that the inner strip corresponds to with a valid stub.

The window in the outer layer channels is what defines the Pt-cut. For a given Pt, the displacement between the hits in the two sensors depends on the position of the module in the tracker, decreasing at larger radii in the case of the barrel. In order to obtain a uniform Pt cut in the tracker volume it is possible to adjust both the sensor spacing during assembly and the width of the coincidence window, which for a single chip is programmable in the range ± 8 strips (symmetrical around the central strip).

Also depending on the position of the strip along the module in the R- Φ plane, the coincidence window must be adjusted to account for the geometrical lateral displacement across the same module as illustrated in Figure 13 a. This offset is independently programmable in the range ± 3 strips in the two halves of the ASIC. Every module is therefore divided into 16 regions of programmable offset.

Table 2

Pt_width register (0001101) settings	coincidence window width
0000xxxx	0 (only central strip)
0001xxxx	central strip ± 1 strip
...	





0011xxxx	central strip ± 3 strips (<u>default</u>)
...	
1010xxxx	central strip ± 8 strips

Table 3

Offset register (0001100) settings for “top” channels 126:2	offset value
xxxxx000	0 (<u>default</u>)
xxxxx001	central strip +1 strip
xxxxx010	central strip +2 strips
xxxxx011	central strip +3 strips
xxxxx101	central strip -1 strip
xxxxx110	central strip -2 strips
xxxxx111	central strip -3 strips

Table 4

Offset register (0001100) settings for “top” channels 254:128	offset value
xx000xxx	0 (<u>default</u>)
xx001xxx	central strip +1 strip
xx010xxx	central strip +2 strips
xx011xxx	central strip +3 strips
xx101xxx	central strip -1 strip
xx110xxx	central strip -2 strips
xx111xxx	central strip -3 strips



3.8.3 Inter-chip boundaries

As described above, both the cluster-width discrimination and the offset correction and correlation logic rely on inputs from neighbouring channels. In particular, the numbers of links required are:

- CWD inner sensor channels: ± 2 adjacent inner-channels;
- CWD outer sensor channels: ± 2 adjacent outer-channels;
- Offset correction: ± 3 adjacent coincidence logic channels;
- Coincidence logic: ± 8 adjacent coincidence logic channels.

In total every coincidence logic channel needs the inputs of the adjacent 30 channels. On both the top and the bottom sides of the ASIC there are therefore 15 input pads and 15 output pads assigned to inter-chip links.

3.8.4 Stubs readout.

When a stub is found, the output of the combinatorial logic for each channel is latched into a parallel-input-serial-output shift register, which is then read out at 40MHz.

The readout is controlled by the circuit in Figure 15.

The readout can be controlled by either the OR_stubs (so the presence of one stub candidate) or by the OR_254 (OR of all the channels, so any activity on chip above threshold). We can select between these two with the control bit SelORin. Then either RawOR or its latched version ORlatched are output offchip to COIN_OR.

ORlatched is used to latch the outputs of the coincidence logic into a 131cell shift register (127channels+1b “tailer”+ 3b header), the readout of which is controlled by a counter and ORlatched.

Table 5

SelORin	RawOR
0	OR_stubs
1	OR_254

Table 6

SelORout	ORchoice
0	ORlatched
1	RawOR

3.9 Pipeline RAM

Figure 16 shows the data path of the CBC2. The *Pipeline RAM* has a depth of 256 bits to give storage capacity of 256 clock cycles (6.4 μ s @ 40MHz) and is 254 bits wide



to match the channel count. The RAM cells are of simple dual-port architecture and are not required to be immune to SEU events since loss of data can be tolerated. *Pipeline Control Logic* sequences writing and reading of data into the *Pipeline RAM*.

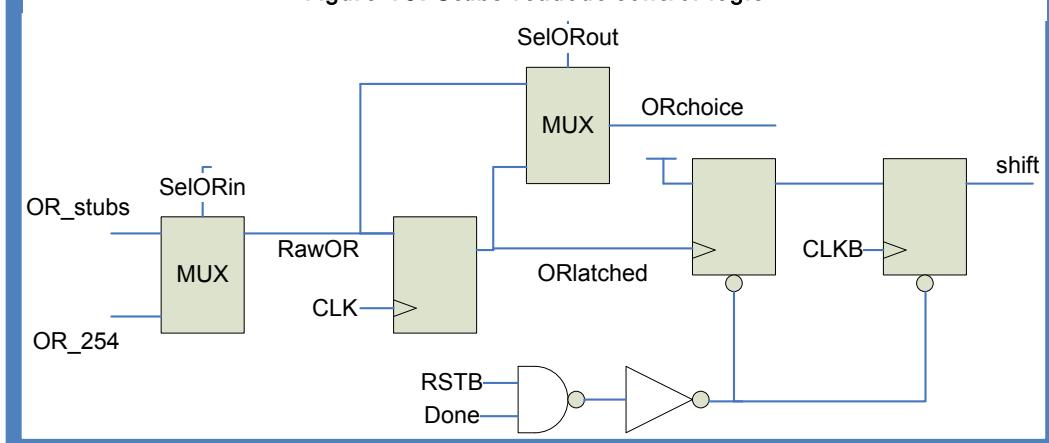
3.10 Data Buffer RAM

The *Data Buffer RAM* stores data from bunch crossings which have been triggered as useful. When this happens data is read from the *Pipeline RAM* and written into the buffer RAM. It has a depth of 32 bits to store 32 events awaiting readout and is 262 bits wide to match the channel count plus the pipeline address from which the data came. The RAM is of the same design as the pipeline RAM and essentially operates as a FIFO. Write and read pointers sequence data to and from the FIFO.

3.11 Data Serialiser

The data stored in the data buffer must be serialised before it can be output from the CBC. This is performed by a 266-to-1 shift register. The 266 bits include the channel data (254), the pipeline address (8), and 2 bit header plus 2 error bits.

Figure 15. Stubs readout control logic



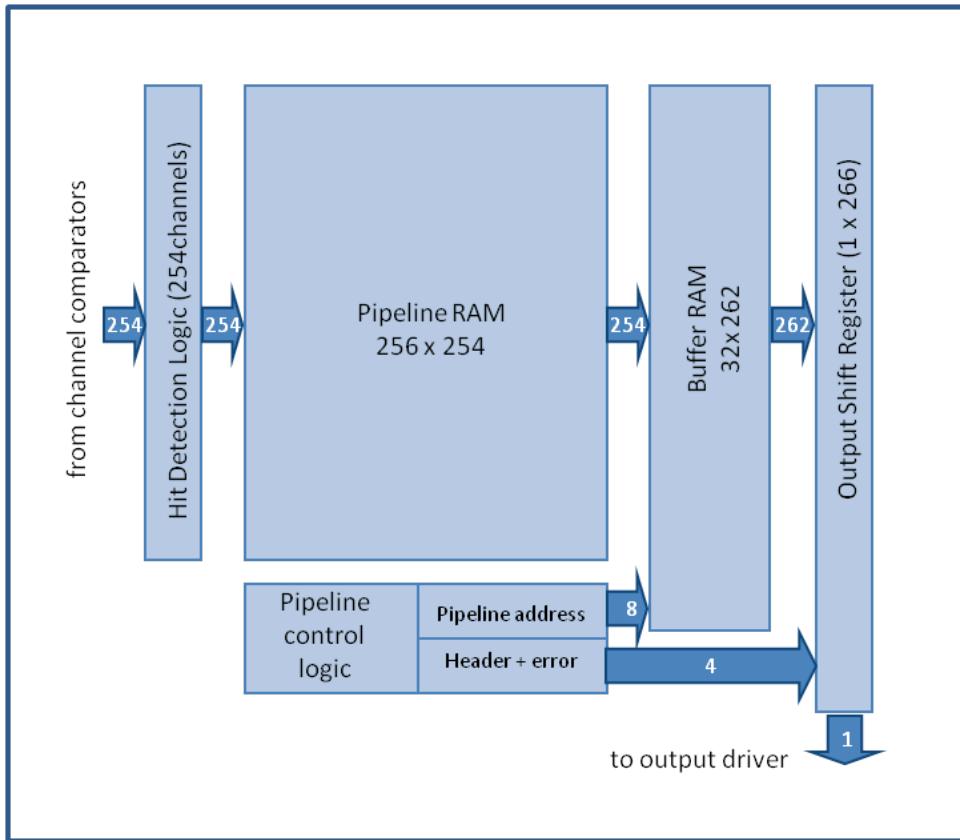


Figure 16. Memory pipeline and readout.

3.12 Pipeline Control Logic

Figure 17 shows a block diagram of the Pipeline *Control Logic*. The sequencing of data being written into the pipeline RAM is controlled by two pointers, the *write pointer*, and the *trigger pointer*. Both of these consist of counters and decoders.

When the CBC2 is first initialised, the *write counter* will start counting from 0. The *write decoder* will convert this count to a RAM position where the data from the front end will be written. The counter will continue counting to 255 before wrapping around to start again. Data previously written will now be overwritten.

The *trigger counter* will not start counting until a predefined latency behind the write counter. The *trigger decoder* will decode this count to point to the RAM position from where data will be read if an external trigger is received.

Pointer Start Logic is used to set the correct latency as determined by the *Latency Register*. The latency between trigger pointer and write pointer is constantly monitored by a latency check circuit. If the measured latency does not match the programmed latency then an error bit is set.

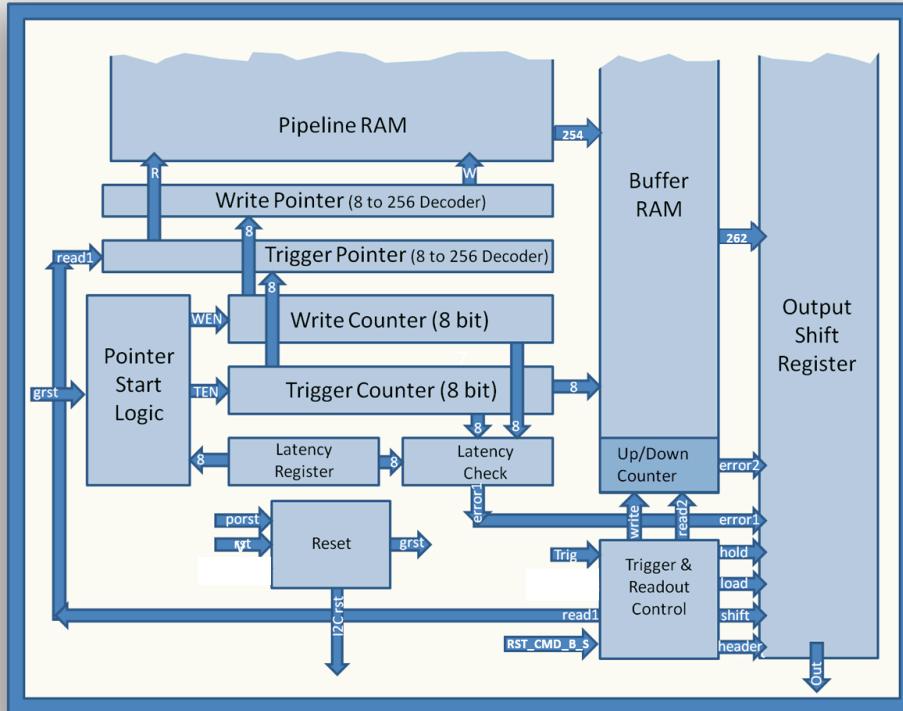


Figure 17. Pipeline Control Logic

The *Reset* circuit takes the chip input reset and the power-on-reset and routes them to the relevant blocks. The power-on-reset (*porst*) and chip input *rst* (*rst*) are used to reset the I₂C and I₂C registers. Both of these resets are used to reset the rest of the logic, but an additional reset, the trigger reset (*RST_CMD_B_S*), is also used. When a *RST_CMD_B_S* is sent it will synchronise all chips to the same clock cycle.

The *Trigger and Readout Control* block has two functions.

Firstly it controls the transfer of data from pipeline to buffer. If a trigger signal is received then a read signal (*read1*) and a buffer write signal (*write*) are generated. These transfer the data and also the address of the pipeline into the buffer.

Secondly, it controls transfer of data from the *Buffer RAM* to the *Output Shift Register*. It monitors whether there is any data in the *Buffer RAM*. If there is none, then it holds the shift register from loading or shifting. Once data becomes available it generates the signals for transferring the data from the *Buffer RAM* to the *Output Shift Register* (*read2*, *load*). The data is then shifted out of the register (*shift*).



A couple of error flags are used to indicate problems with the operation of the circuit. The first is set by the *Latency Check* circuit. This monitors the difference between the *Write Counter* and *Trigger Counter* and compares it to the *Latency Register*. If there is a difference, the latency error flag is set. The second error comes from the *up-down counter*. This holds the number of items stored in the *Buffer RAM* and if this reaches 32 then a FIFO full flag is set. Both of these error flags are read out in the header of each data packet.

3.13 Slow Control

The *Slow Control* is the interface between the external system and the control registers which set up how the CBC2 will operate. It is implemented in I²C slave architecture.

Figure 18 shows a typical I²C configuration, with one master controlling several slaves. In addition to sending and receiving data (SDA), the master also generates the clock (SCL). I²C output drivers are open-drain, requiring pull-up resistors. The maximum value of resistor that can be used is determined by the clock frequency and total capacitance on the I²C bus. In addition it must be remembered that for small values of resistor, the open-drain outputs will not be able to pull the signals all the way down to 0V.

Since all devices share a common bus, only one may use it at a time.

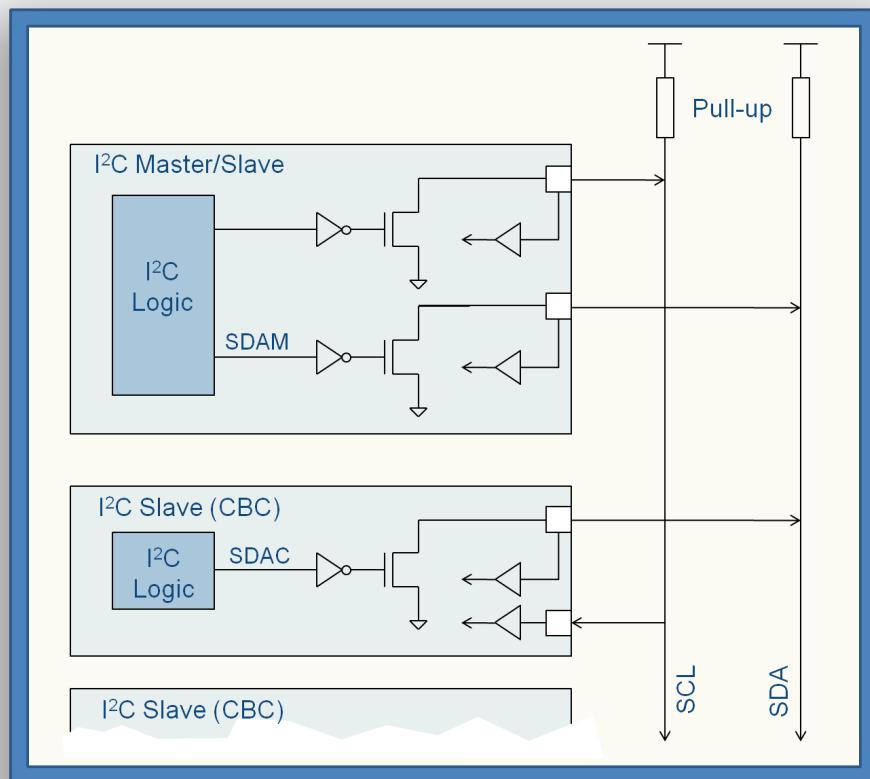


Figure 18. I²C Master/Slave Configuration

The CBC2 chip has 5 address inputs, which compose the 5 LSBs of the address. I²C chip addresses must consist of 7 bits, so the remaining CBC2 address bits have been internally wired to “10”. Therefore, when sending an I²C command to an individual chip, the 2 most significant bits of the chip address must be set as “10”. For example, to address a chip with the 5 address inputs set to “00000”, the I²C command must have a chip address of “1000000”.

Figure 19 shows the signals for an I²C “write” transaction. When the I²C is inactive, both the clock and data buses are high. A transaction must start with an I²C write condition. This occurs when the data line (SDA) goes low when the clock (SCL) is high. This is initiated by the master on SDAM. During this time the slave is inactive (SDAC). All data must then change only when the clock is low.

The master sends the address of the slave it wants to communicate with, which in this case is 1000001, followed by a “0” which indicates it wants to write information to the slave. It then releases SDA by taking SDAM high and the slave acknowledges receipt of the address by taking SDAC (and hence SDA) low.



The master then sends the address of the register it wants to write to (in this case 00000001), and the slave acknowledges. The last piece of information is the data that is to be written into the register (00001110). The slave acknowledges, and the master sends the I²C stop condition (SDA goes high when clock is high).

Every register on any chip can be written to in this way by supplying the relevant chip and register address. The chip address “1111111” is reserved for addressing all chips connected to the same bus at the same time.

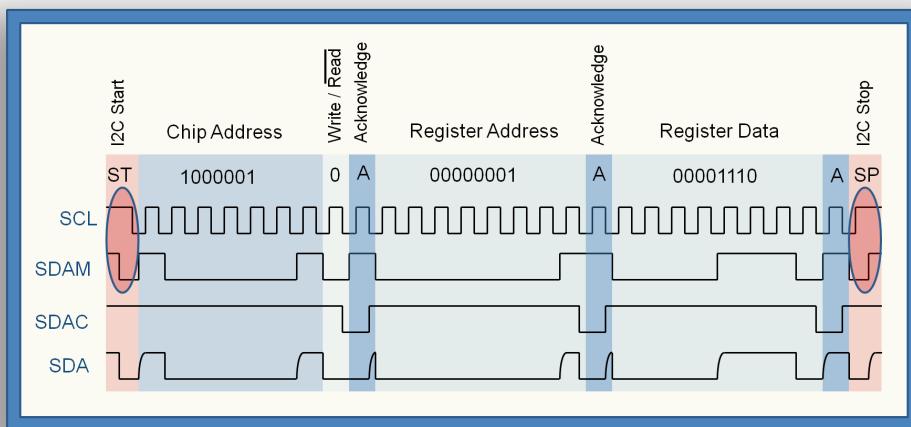


Figure 19. I²C Write Transaction

Figure 20 shows the format for read from a register as well as writing to it. As the write operation has already been described this section will deal with the read transaction.

To begin a read transaction, the register on the chip you want to read from has to be addressed. This is done by firstly sending a start condition and addressing the chip with the write/read bit set to 0 (write mode). Then the register that is to be read is addressed and a stop condition is sent. This has set up the internal I²C interface to address the relevant register.

Secondly, beginning with a start condition, the chip has to be addressed for a second time, but in this case, the write/read bit is set to 1 to indicate a read transaction. The register data is then loaded onto an internal bus, the master releases the SDA bus and the slave, detecting a read condition, outputs the register data over the SDA bus. The slave releases the bus, and the master acknowledges.



All control register are written to and read from using the I2C interface. The next sections list all available control registers.

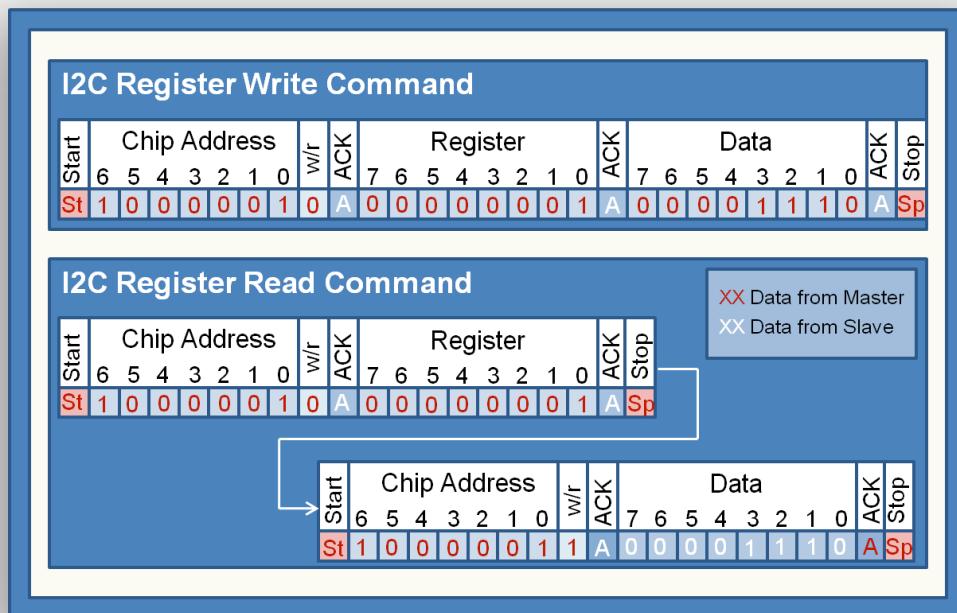


Figure 20. I2C Write/Read Transaction



3.14 Control Registers

The control registers define how the CBC2 will operate. On start-up all registers will reset to default values. A full list of register is in the section “**CBC2 I2C Registers**”.

Table 7. Front End Control (I2C Register Address 00000000)

Bit	Function	Default
7	“page” bit (0 writes to page 1)	0
6	Comparator Polarity (0=holes, 1=electrons)	0
5	Comparator Hysteresis Bit 3	1
4	Comparator Hysteresis Bit 2	1
3	Comparator Hysteresis Bit 1	1
2	Comparator Hysteresis Bit 0	1
1	Postamp Polarity (0=holes, 1=electrons)	0
0	Preamp Polarity (0=holes, 1=electrons)	0

Table 7 shows front end control bits. The polarity of the signal is set by bits 0,1 & 6. For electron polarity these are set to 1, for holes these are set to 0. The comparator hysteresis is set using bits 2-5. Maximum hysteresis occurs when all four bits are set to 1.

The defaults settings for the front end control register are for holes polarity and for maximum hysteresis. The register address is 00000000.

Table 8. Trigger Latency (I2C Register Address 00000001)

Bi t	Function	Default
7	Trigger Latency Bit 7	1
6	Trigger Latency Bit 6	1
5	Trigger Latency Bit 5	0
4	Trigger Latency Bit 4	0



3	Trigger Latency Bit 3	1
2	Trigger Latency Bit 2	0
1	Trigger Latency Bit 1	0
0	Trigger Latency Bit 0	0

Table 8 shows the trigger latency register. This defines the pipeline separation in clock cycles between the write and trigger pointers. The default value on power up is 200 clocks. The register address is 00000001

Table 9. Hit Detect & SLVS (I2C Register Address 00000010)

Bi t	Function	Default
7	Enable 160MHz SLVS pads (Default is OFF)	1
6	Hit Detect Mode (0=single, 1=variable)	0
5	Hit Detect Enable (0=OFF, 1=ON)	1
4	SLVS Off (0=ON, 1=OFF)	0
3	SLVS Current Bit3	1
2	SLVS Current Bit2	0
1	SLVS Current Bit 1	0
0	SLVS Current Bit 0	0

Table 9 Shows the Hit Detect and SLVS control register. The hit detect modes are controlled by bits 5 and 6. Bit 6 controls the mode of operation – whether it outputs a pulse of a single clock cycle or a pulse of variable length. The default is 0 which is for a single pulse. Bit enable/disables the hit detect logic. The default value is 1 which is ON.

Bit 4 is the disable for the SLVS receiver circuit. The default is 0, which is ON. Bits 0-3 control the current to the SLVS transmitter. The bits are active low which means the minimum current (0mA) is set by “1111”. The default value is for a current of 2mA and the maximum setting “0000” gives a current of about 2.5mA. The address of this register is 00000010



Table 10. Channel Offset Registers

Channel	I2C Address	Default
1	00000001	10000000
2	00000010	10000000
etc	etc	10000000
253	11111101	10000000
254	11111110	10000000
Dummy	11111111	10000000

Table 10 shows the channel offset registers. The values written into these registers control the amount of offset on the output of the postamplifiers. Values of 00000000 give 0V offset . The maximum offset is given by a setting of 1111111. The default setting is for a mid-range offset. The register addresses are 00000001 to 11111111 on address page2.

With default bias settings, the offset range is 0-200mV with the default set to 100mV. However this range can be programmed by changing the Ipaos bias setting (see Bias Generator).

3.15 Bias Generator

The *bias generator* provides all of the bias currents and voltages necessary for operation of the front end analogue circuitry. Each current and voltage is set by loading a control register as listed in Table 11. The digital output from these registers is then converted into the relevant bias by several DACs.

Table 11. Bias Generator Registers

Name	Function	Nominal	Range	I2C Address
Ipre1	Preamp Input Branch Bias Current	90uA	0-255uA	00000011
Ipre2	Preamp Cascode Branch Bias Current	10uA	0-51uA	00000100
Ipsf	Preamp Source Follower Bias Current	25uA	0-51uA	00000101
Ipa	Postamp Bias Current	20uA	0-51uA	00000110
Ipaos	Postamp Offset Adjust Bias Current	4uA	0-12.7uA	00000111
VPAFB	Postamp Feedback Bias Current	2.5uA	0-25.5uA	00001000
Icomp	Comparator Bias Current	2uA	0-12.7uA	00001001
Vpc	Preamp Cascode Bias Voltage	0.4V	0.2-0.8V	00001010



Vplus	Postamp Bias Voltage	0.6V	0.2-0.8V	00001011
VCth	Comparator Threshold Voltage	0.6V	0.2-0.8V	00001100
CAL_I	Calibration Circuit Bias Current	5uA	0-12uA	00010000
CAL_VCAS_C	Calibration Circuit Cascode Bias	0.4V*	0.2-0.8V	00010001

*NB: nominal value =0.4V differs from default value =0.2V

The bias generator requires a master reference current, which is provided by an on-chip bandgap reference.

3.16 LDO Voltage Regulator

An LDO Voltage Regulator is included on chip as an independent block. All connections to the rest of the ASIC must be applied externally. The Input to the LDO should be 1.2V and can be the same supply as to the digital circuitry (VDDD). The output from the LDO can be connected to VDDA to supply the analogue circuitry. External capacitance must be added to the output to avoid oscillation.

3.17 SLVS I/O

Two test pads (160M_DIFF_OUT_N/P, 160M_DIFF_IN_N/P) and the clock use SLVS I/O (Scalable Low Voltage Signalling). The transmitter has programmable current with default setting of 2mA. With 100Ω termination at the receiving end this gives a differential voltage of 200mV centred on about 1.25V (Figure 21).

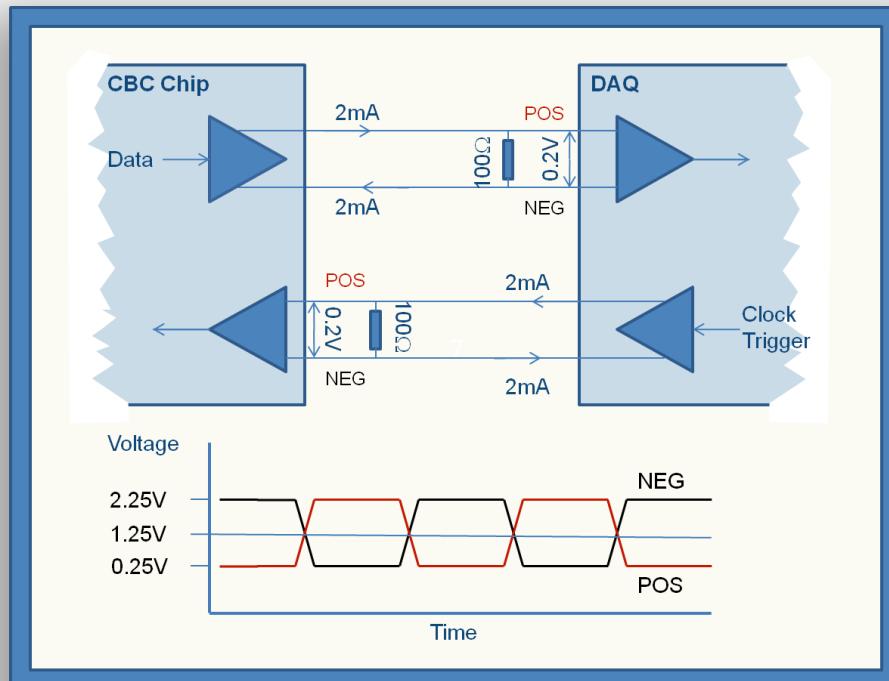


Figure 21. SLVS Data Format.



3.18 DC-DC Converter

A *DC-DC converter* has been placed as a completely independent block on the ASIC. This block can be used to generate The 1.25V supply from a 2.5V supply. There are no connections to the rest of the ASIC on the chip. All connections will have to be made externally.

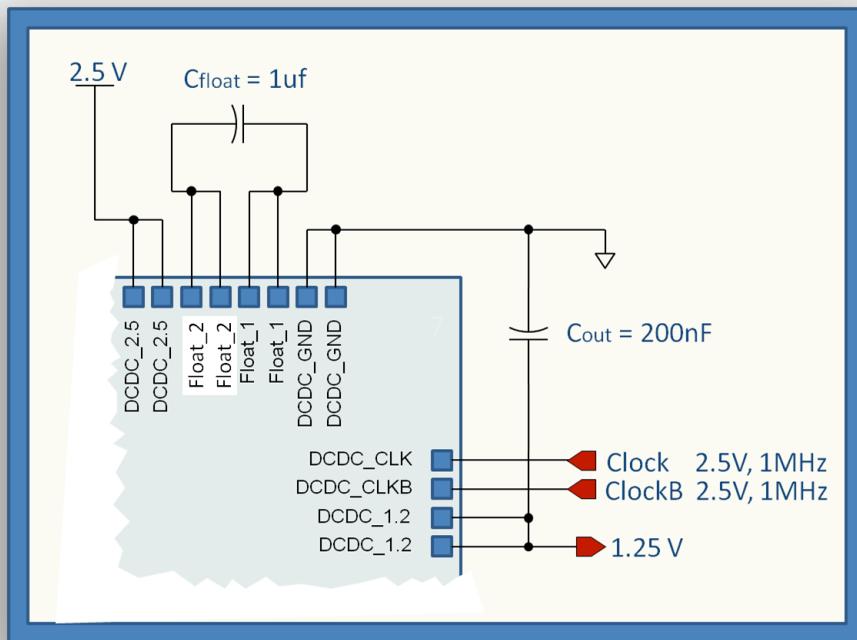
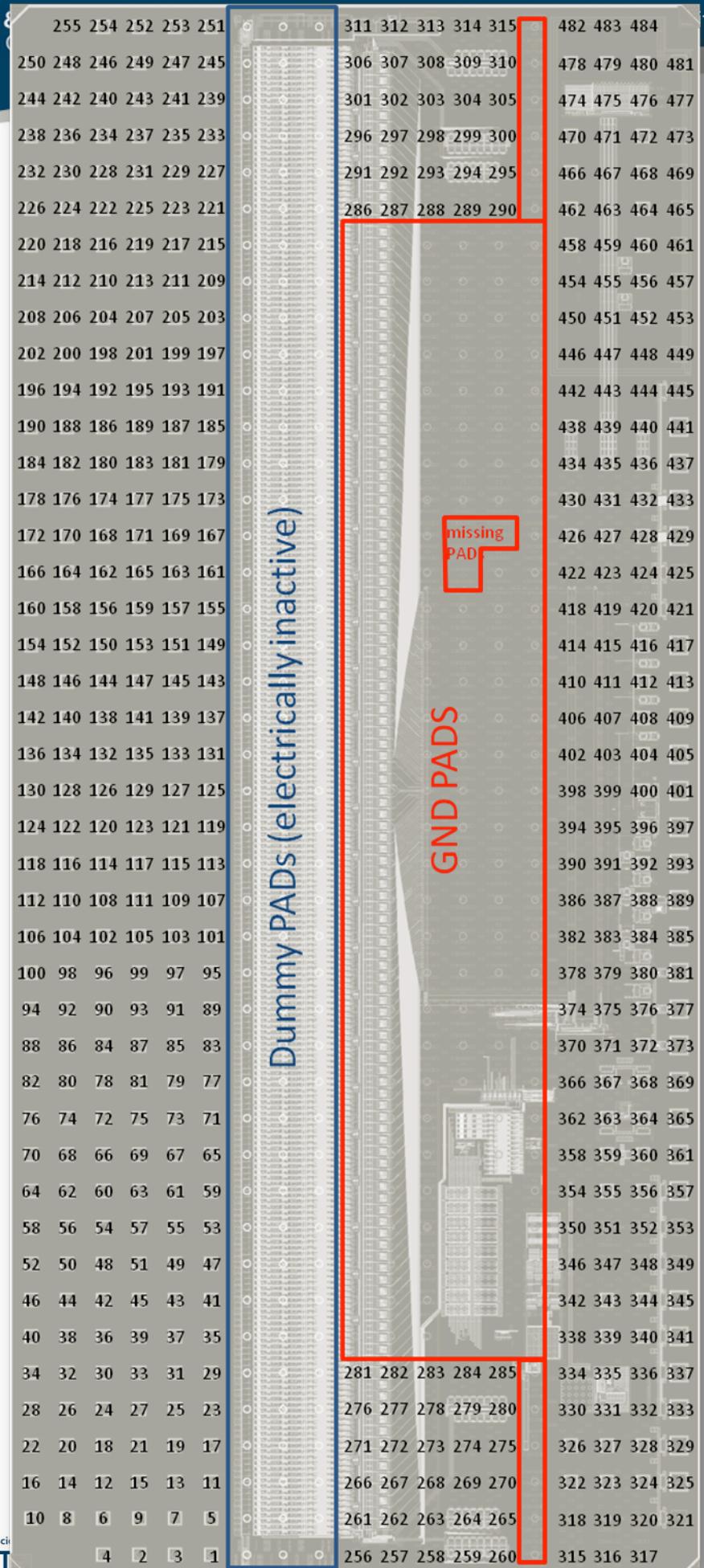


Figure 22. Connections for DC-DC Converter (PAD positions not accurate).

The *DC-DC Converter* should be connected as shown in Figure 22. A 1uF capacitor (C_{float}) is connected between pads *Float_2* and *Float_1*. A 200nF capacitor (C_{out}) is connected between the output (1.25V) and ground. The circuit is clocked using the *Clock* input, a 2.5V signal running at 1MHz. Since the clock signal may cause interference to the front end of the chip by coupling through stray capacitance, a dummy clock signal (*ClockB*) is provided for balance.







4. CBC2 Input/Output Pads

Figure 23 shows the position of the pads on the CBC2 chip. The PADs are C4 (Controlled Collapse Chip Connection) 5on10 (125 μ m ball diameter, 250 μ m pitch), except the last column to the right of Figure 23, which are probeable PADs for wafer testing. The following tables list the functions of each pad.

Table 12. Input PADs

Pad	Name	Type	Function
1	IN<1>	Analogue	Bottom sensor: input 1 (first)
2	IN<2>	Analogue	Top sensor: input 1 (first)
3	IN<3>	Analogue	Bottom sensor: input 2
4	IN<4>	Analogue	Top sensor: input 2
...			
2n	IN<2n-1>	Analogue	Bottom sensor: input n
2n+1	IN<2n>	Analogue	Top sensor: input n
...			
253	IN<253>	Analogue	Bottom sensor: input 127 (last)
254	IN<254>	Analogue	Top sensor: input 127 (last)
255	UNCON<1>	-	unconnected PAD (has ESD protection)

Table 13. PADs connecting previous chip.

PAD	Name	Type	Function
256	IN_A<126>	1.2V digital input	input from previous chip's A<126> (output of CWD for bottom sensor's channel 126 --> channel 251)
257	IN_B<127>	"	input from previous chip's B<127> (output of CWD for top sensor's channel 127 --> channel 254)
258	IN_D<119>	"	input from previous chip's D<119> (output of coincidence logic for bottom sensor's channel 119 --> channel 237)
259	IN_D<122>	"	input from previous chip's D<122> (output of coincidence logic for bottom sensor's channel 122 --> channel 243)
260	IN_D<125>	"	input from previous chip's D<125> (output of coincidence logic for bottom sensor's channel 125 --> channel 249)
261	IN_A<127>	"	input from previous chip's A<127> (output of CWD for bottom sensor's channel 127 --> channel 253)
262	IN_D<117>	"	input from previous chip's D<117> (output of coincidence logic for bottom sensor's channel 117 --> channel 233)
263	IN_D<120>	"	input from previous chip's D<120> (output of coincidence logic for bottom sensor's channel 120 --> channel 239)
264	IN_D<123>	"	input from previous chip's D<123> (output of coincidence logic for bottom sensor's channel 123 --> channel 245)
265	IN_D<126>	"	input from previous chip's D<126> (output of coincidence logic for bottom sensor's channel 126 --> channel 251)
266	IN_B<126>	"	input from previous chip's B<126> (output of CWD for top sensor's channel 126 --> channel 252)
267	IN_D<118>	"	input from previous chip's D<118> (output of coincidence logic for bottom sensor's channel 118 --> channel 235)
268	IN_D<121>	"	input from previous chip's D<121> (output of coincidence logic for bottom sensor's channel 121 --> channel 241)
269	IN_D<124>	"	input from previous chip's D<124> (output of coincidence



			logic for bottom sensor's channel 124 --> channel 247)
270	IN_D<127>	"	input from previous chip's D<127> (output of coincidence logic for bottom sensor's channel 127 --> channel 253)
271	OUT_A<1>	1.2V digital output	output to previous chip's IN_A<1> (output of CWD for bottom sensor's channel 1 --> channel 1)
272	OUT_B<2>	"	output to previous chip's IN_B<2> (output of CWD for bottom sensor's channel 4 --> channel 4)
273	OUT_D<3>	"	output to previous chip's IN_D<3> (output of coincidence logic for bottom sensor's channel 3 --> channel 5)
274	OUT_D<6>	"	output to previous chip's IN_D<6> (output of coincidence logic for bottom sensor's channel 6 --> channel 11)
275	OUT_D<9>	"	output to previous chip's IN_D<9> (output of coincidence logic for bottom sensor's channel 9 --> channel 17)
276	OUT_A<2>	"	output to previous chip's IN_A<2> (output of CWD for bottom sensor's channel 2 --> channel 2)
277	OUT_D<1>	"	output to previous chip's IN_D<1> (output of coincidence logic for bottom sensor's channel 1 --> channel 1)
278	OUT_D<4>	"	output to previous chip's IN_D<4> (output of coincidence logic for bottom sensor's channel 4 --> channel 7)
279	OUT_D<7>	"	output to previous chip's IN_D<7> (output of coincidence logic for bottom sensor's channel 7 --> channel 13)
280	OUT_D<10>	"	output to previous chip's IN_D<10> (output of coincidence logic for bottom sensor's channel 10 --> channel 19)
281	OUT_B<1>	"	output to previous chip's IN_B<1> (output of CWD for bottom sensor's channel 1 --> channel 2)
282	OUT_D<2>	"	output to previous chip's IN_D<2> (output of coincidence logic for bottom sensor's channel 2 --> channel 3)
283	OUT_D<5>	"	output to previous chip's IN_D<5> (output of coincidence logic for bottom sensor's channel 5 --> channel 9)
284	OUT_D<8>	"	output to previous chip's IN_D<8> (output of coincidence logic for bottom sensor's channel 8 --> channel 15)
285	OUT_D<11>	"	output to previous chip's IN_D<11> (output of coincidence logic for bottom sensor's channel 11 --> channel 21)

Table 14. PADs connecting next chip.

PAD	Name	Type	Function
286	IN_B<1>	1.2V digital input	input from next chip's B<1> (output of CWD for top sensor's channel 1 --> channel 2)
287	IN_D<2>	"	input from next chip's D<2> (output of coincidence logic for bottom sensor's channel 2 --> channel 3)
288	IN_D<5>	"	
289	IN_D<8>	"	
290	IN_D<11>	"	
291	IN_A<2>	"	input from next chip's A<2> (output of CWD for bottom sensor's channel 2 --> channel 3)
292	IN_D<1>	"	
293	IN_D<4>	"	
294	IN_D<7>	"	
295	IN_D<10>	"	
296	IN_A<1>	"	
297	IN_B<2>	"	
298	IN_D<3>	"	
299	IN_D<6>	"	
300	IN_D<9>	"	



301	OUT_B<126>	1.2V digital output	output to next chip's IN_B<126> (output of CWD for top sensor's channel 126 --> channel 254)
302	OUT_D<118>	"	output to previous chip's IN_D<118> (output of coincidence logic for bottom sensor's channel 118 --> channel 235)
303	OUT_D<121>	"	
304	OUT_D<124>	"	
305	OUT_D<127>	"	
306	OUT_A<127>	"	output to next chip's IN_A<127> (output of CWD for bottom sensor's channel 127 --> channel 253)
307	OUT_D<117>	"	
308	OUT_D<120>	"	
309	OUT_D<123>	"	
310	OUT_D<126>	"	
311	OUT_A<126>	"	
312	OUT_B<127>	"	
313	OUT_D<119>	"	
314	OUT_D<122>	"	
315	OUT_D<125>	"	

Table 15. Back-end PADs.

PAD	Name	Type	Function
315:321	GND	Supply	
322:329	VLDOI	Supply	LDO input (1.2V)
330:337	VLDOO	Supply	LDO output (~1.1V)
338:345	VDDA	Supply	VDD Analog (~1.1V)
346:353	GND	Supply	
354:357	AMUX_OUT	Analogue	Analogue-mux output
358:361	160M_SE_OUT	SLVS digital	high-speed, single-ended driver output
362	160M_DIFF_OUT_N	SLVS digital	high-speed, SLVS, differential driver output - negative terminal
363:365	160M_DIFF_OUT_P	SLVS digital	high-speed, SLVS, differential driver output - positive terminal
366	160M_DIFF_IN_N	SLVS digital	high-speed, SLVS, differential driver input - negative terminal
367:369	160M_DIFF_IN_P	SLVS digital	high-speed, SLVS, differential driver input - positive terminal
370	GND	Supply	
371:373	COIN_OR	1.2V digital	coincidence OR (fastOR) output
374	GND	Supply	
375:377	COIN_DATA_OUT	1.2V digital	coincidence data output
378	ADD<0>	1.2V digital	I2C address <0>
379:381	DATA_OUT	1.2V	data output



		digital	
382	ADD<1>	1.2V digital	I2C address <1>
383:385	RESET	1.2V digital	Chip reset
386	ADD<2>	1.2V digital	I2C address <2>
387:389	SDA	Open drain	I2C Data line
390	ADD<3>	1.2V digital	I2C address <3>
391:393	SCLK	1.2V digital	I2C clock line
394	ADD<4>	1.2V digital	I2C address <4>
395:397	CLK_40_N	SLVS digital	40MHz clk (negative)
398	GND	Supply	
399:401	CLK_40_P	SLVS digital	40MHz clk (positive)
402	GND	Supply	
403:405	I2C_REFRESH	1.2V digital	I2C registers refresh
406	GND	Supply	
407:409	TEST_PULSE	1.2V digital	Test pulse request
410	GND	Supply	
411:413	FAST_RESET	1.2V digital	(this used to be the Rst101 in CBC1)
414	GND	Supply	
415:417	T1_TRIGGER	1.2V digital	Readout trigger
418:425	GND	Supply	
426:433	VDDD	Supply	
434:441	DCDC_1_2	Supply	DCDC Output
442:445	GND	Supply	
446:449	DCDC_GND	Supply	
450	DCDC_GND	Supply	
451:453	DCDC_CLK_N	2.5V digital	DCDC clock input negative (Dummy, connected only to ESD protection)
454	DCDC_GND	Supply	
455:457	DCDC_CLK_P	2.5V digital	DCDC clock input positive
458:465	FLOAT_1	Analogue	DCDC float capacitance (was FLOAT_1 in CBC1)
466:473	FLOAT_2	Analogue	DCDC float capacitance (was FLOAT_0 in CBC1)
474:477	DCDC_GND	Supply	
478:484	DCDC_2_5	Supply	DCDC input (2.5V)



5. Data Output Format

The data from the CBC2 is purely digital and will be output at the clock frequency (40MHz). The data will consist of a digital header (2 logic 1 bits), 2 error bits, the pipeline address (8 bits), and the data from all 254 channels (Figure 24). The full data stream is therefore 266 bits in length.

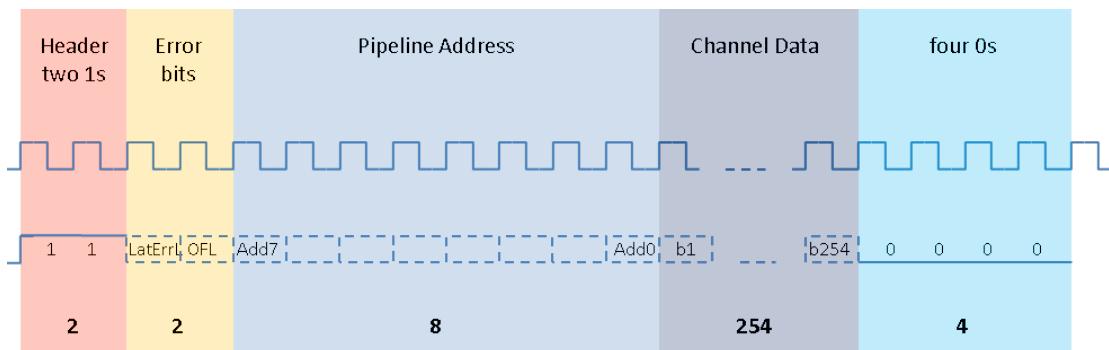


Figure 24. Data Format

The timing between the first trigger being received and the first data packet is shown in Figure 25. There is a 5 clock cycle delay between the external trigger signal and the data being output (2 fewer than in CBC1).

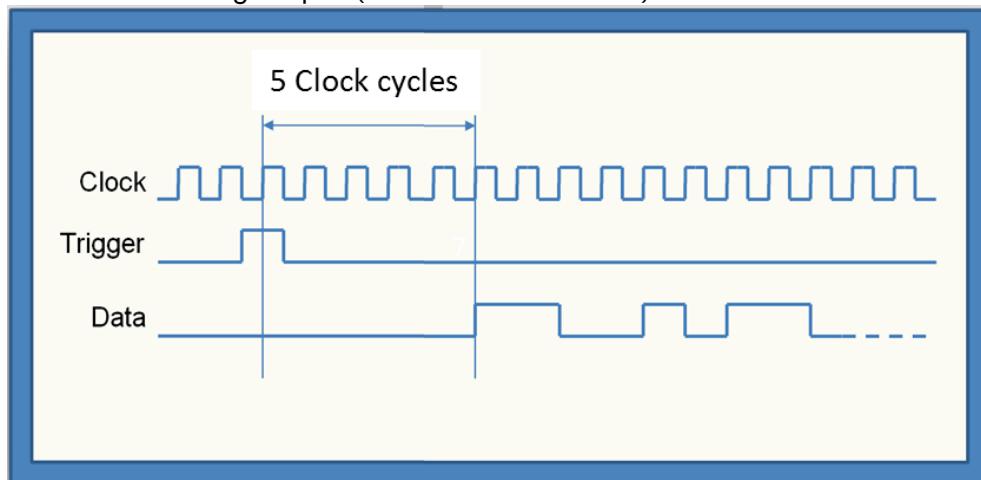


Figure 25. Data Timing



CBC2 I2C Registers

Author: M.L. Prydderch/P.Murray

Issue: 1.0.

Published: 01/10/12

NOTE: This document is based on an original document by Peter Murray entitled "CBC2 chip calibration system". This document supersedes the latter.



Contents

Tables	42
General description.....	44
Register Bit Assignment.....	44

Tables

Table 1 -- Front End Control: Address 00000000	44
Table 2 -- Trigger Latency: Page 1 Address 00000001	45
Table 3 -- Hit Detect and SLVS : Page 1 Address 00000010	45
Table 4 -- Ipre1: Page 1 Address 00000011.....	45
Table 5 -- Ipre2: Page1 Address 00000100.....	46
Table 6 -- Ipsf: Page 1 Address 00000101	46
Table 7 -- Ipa: Page 1 Address 00000110	46
Table 8 -- Ipaos: Page 1 Address 00000111	47
Table 9 -- VPAFB: Page 1 Address 00001000	47
Table 10 -- Icomp: Page 1 Address 00001001	47
Table 11 -- Vpc: Page 1 Address 00001010.....	48
Table 12 -- Vplus: Page 1 Address 00001011	48
Table 13 -- VCth: Page 1 Address 00001100	48
Table 14 -- Test Pulse Potentiometer Node Select: Page 1 Address 00001101	49
Table 15 – Select of Test Pulse Delay & Test Channel Group: Page 1 Address 00001110	49
Table 16 -- Misc Test Pulse Control & Analogue Mux: Page 1 Address 00001111.....	50
Table 17 -- Test Pulse Charge Pump Reference Current: Page 1 Address 00010000	50



Table 18 -- Test Pulse Charge Pump Mirror Cascode Voltage: Page 1 Address 00010001	50
Table 19 -- CWD Window and Coincidence Offset: Page 1 Address 00011000	52
Table 20 -- Miscellaneous Stub Logic: Page 1 Address 00011001	52
Table 21 -- Mask Channel<1:8> up to <241:248>: Page 1 Address 00100000 up to 00111110	52
Table 22 -- Mask Channel<249:256>: Page 1 Address 00111111	53
Table 23 -- Channel Offset Word<1> up to <253>: Page 2 Address 00000001 up to 11111101	53
Table 24 -- Channel Offset Word<254>: Page 2 Address 11111110	53
Table 25 -- Channel Offset Dummy Word: Page 2 Address 11111111	54



General description

The CBC2 chip is configured by programming a series of 8 bit registers via the I2C interface. The chip requires more than the 256 register addresses available with an 8 bit address, so a paging system is employed to increase the address space. With this system addresses can be used twice with the page bit determining which of two registers is actually being addressed. Register address 0 is considered page free so that it can always be written to. This register includes a bit location (bit 7) that can be set to 1 or 0 to determine which of the two pages are being addressed. The default is 0 for page 1 which includes all of the main bias configuration registers. This system gives an address space with 511 addresses.

The page set by writing into register address 0 will remain active until it is changed by another write to register 0. A reset will return register address 0 to its default condition.

Register Bit Assignment

The following tables provide a bit-by-bit allocation for each register.

Note: Bit 7 is the first to be loaded in time sequence during I2C transfer. This is usually the MSB where groups of bits form command words.

Table 16 -- Front End Control: Address 00000000

Bit	Function	Default
7	“page” bit (0 writes to page 1)	0
6	Comparator polarity (0=holes, 1=electrons)	0
5	Comparator hysteresis Bit 3 MSB	1
4	Comparator hysteresis Bit 2	1
3	Comparator hysteresis Bit 1	1
2	Comparator hysteresis Bit 0 LSB	1
1	Postamp Polarity (polarity (0=holes, 1=electrons)	0
0	Preamp Polarity (polarity (0=holes, 1=electrons)	0

**Table 17 -- Trigger Latency: Page 1 Address 00000001**

Bit	Function	Default
7	Trigger Latency bit 7 MSB	1
6	Trigger Latency bit 6	1
5	Trigger Latency bit 5	0
4	Trigger Latency bit 4	0
3	Trigger Latency bit 3	1
2	Trigger Latency bit 2	0
1	Trigger Latency bit 1	0
0	Trigger Latency bit 0 LSB	0

Table 18 -- Hit Detect and SLVS : Page 1 Address 00000010

Bit	Function	Default
7	Enable 160MHz SLVS pads (Default is OFF)	1
6	Hit Detect Mode (0=single, 1=variable)	0
5	Hit Detect enable (0=OFF, 1=ON)	1
4	SLVS Off (0=OFF, 1=ON)	0
3	SLVS Current bit 3 MSB	1
2	SLVS Current bit 2	0
1	SLVS Current bit 1	0
0	SLVS Current bit 0 LSB	0

Table 19 -- Ipre1: Page 1 Address 00000011

Bit	Function	Default
7	Preamp input Branch Current (MSB)	0
6	Preamp input Branch Current	1
5	Preamp input Branch Current	0
4	Preamp input Branch Current	0
3	Preamp input Branch Current	0
2	Preamp input Branch Current	1
1	Preamp input Branch Current	1
0	Preamp input Branch Current (LSB)	0

**Table 20 -- Ipre2: Page1 Address 00000100**

Bit	Function	Default
7	Preamp Cascode Branch Current (MSB)	0
6	Preamp Cascode Branch Current	0
5	Preamp Cascode Branch Current	1
4	Preamp Cascode Branch Current	0
3	Preamp Cascode Branch Current	1
2	Preamp Cascode Branch Current	1
1	Preamp Cascode Branch Current	1
0	Preamp Cascode Branch Current (MSB)	0

Table 21 -- Ipsf: Page 1 Address 00000101

Bit	Function	Default
7	Preamp Source Follower Bias Current (MSB)	0
6	Preamp Source Follower Bias Current	1
5	Preamp Source Follower Bias Current	1
4	Preamp Source Follower Bias Current	1
3	Preamp Source Follower Bias Current	1
2	Preamp Source Follower Bias Current	0
1	Preamp Source Follower Bias Current	1
0	Preamp Source Follower Bias Current (LSB)	0

Table 22 -- Ipa: Page 1 Address 00000110

Bit	Function	Default
7	Postamp Bias Current (MSB)	0
6	Postamp Bias Current	1
5	Postamp Bias Current	1
4	Postamp Bias Current	0
3	Postamp Bias Current	1
2	Postamp Bias Current	0
1	Postamp Bias Current	1
0	Postamp Bias Current (LSB)	0

**Table 23 -- Ipaos: Page 1 Address 00000111**

Bit	Function	Default
7	Postamp Offset adjust Bias Current (MSB)	0
6	Postamp Offset adjust Bias Current	1
5	Postamp Offset adjust Bias Current	0
4	Postamp Offset adjust Bias Current	0
3	Postamp Offset adjust Bias Current	1
2	Postamp Offset adjust Bias Current	0
1	Postamp Offset adjust Bias Current	1
0	Postamp Offset adjust Bias Current (LSB)	1

Table 24 -- VPAFB: Page 1 Address 00001000

Bit	Function	Default
7	Postamp Feedback Bias Current (MSB)	0
6	Postamp Feedback Bias Current	1
5	Postamp Feedback Bias Current	1
4	Postamp Feedback Bias Current	0
3	Postamp Feedback Bias Current	0
2	Postamp Feedback Bias Current	0
1	Postamp Feedback Bias Current	0
0	Postamp Feedback Bias Current (LSB)	0

Table 25 -- Icomp: Page 1 Address 00001001

Bit	Function	Default
7	Comparator Bias Current (MSB)	0
6	Comparator Bias Current	0
5	Comparator Bias Current	1
4	Comparator Bias Current	0
3	Comparator Bias Current	0
2	Comparator Bias Current	0
1	Comparator Bias Current	1
0	Comparator Bias Current (LSB)	1

**Table 26 -- Vpc: Page 1 Address 00001010**

Bit	Function	Default
7	Preamp Cascode Bias Voltage (MSB)	0
6	Preamp Cascode Bias Voltage	0
5	Preamp Cascode Bias Voltage	1
4	Preamp Cascode Bias Voltage	1
3	Preamp Cascode Bias Voltage	1
2	Preamp Cascode Bias Voltage	1
1	Preamp Cascode Bias Voltage	1
0	Preamp Cascode Bias Voltage	1

Table 27 -- Vplus: Page 1 Address 00001011

Bit	Function	Default
7	Postamp Bias Voltage (MSB)	1
6	Postamp Bias Voltage	0
5	Postamp Bias Voltage	0
4	Postamp Bias Voltage	0
3	Postamp Bias Voltage	0
2	Postamp Bias Voltage	0
1	Postamp Bias Voltage	0
0	Postamp Bias Voltage (LSB)	0

Table 28 -- VCth: Page 1 Address 00001100

Bit	Function	Default
7	Comparator Threshold Voltage (MSB)	0
6	Comparator Threshold Voltage	1
5	Comparator Threshold Voltage	1
4	Comparator Threshold Voltage	1
3	Comparator Threshold Voltage	1
2	Comparator Threshold Voltage	1
1	Comparator Threshold Voltage	1
0	Comparator Threshold Voltage (LSB)	1

**Table 29 -- Test Pulse Potentiometer Node Select: Page 1 Address
00001101**

Bit	Function	Default
7	Test Pulse Potentiometer Node Select (MSB)	0
6	Test Pulse Potentiometer Node Select	0
5	Test Pulse Potentiometer Node Select	0
4	Test Pulse Potentiometer Node Select	0
3	Test Pulse Potentiometer Node Select	0
2	Test Pulse Potentiometer Node Select	0
1	Test Pulse Potentiometer Node Select	0
0	Test Pulse Potentiometer Node Select (LSB)	0

Note – default settings give pot output = 1.1V. Setting bit 7 to 1 gives half range

**Table 30 – Select of Test Pulse Delay & Test Channel Group: Page 1
Address 00001110**

Bit	Function	Default
7	Test Delay Select (LSB)	0
6	Test Delay Select	0
5	Test Delay Select	0
4	Test Delay Select	0
3	Test Delay Select (MSB)	0
2	Select Channel Group To Test (LSB)	0
1	Select Channel Group To Test	0
0	Select Channel Group To Test (MSB)	0

Note: Test Pulse Delay select 00000 gives a unit delay and 11001 gives the longest delay. There are no delays corresponding to the remaining codes – the circuit outputs no pulse

Note: Select Channel Group To Test 000 gives lowest numbered channel group.



Table 31 -- Misc Test Pulse Control & Analogue Mux: Page 1 Address 00001111

Bit	Function	Default
7	Test Pulse Polarity Select (= 1 for positive edge)	0
6	Enable Test Pulse (= 1 to enable)	0
5	Ground the Test Pulse capacitor plates of non selected Test Pulse groups (=1 to ground capacitors)	0
4	Analogue Mux Select <4>	0
3	Analogue Mux Select <3>	0
2	Analogue Mux Select <2>	0
1	Analogue Mux Select <1>	0
0	Analogue Mux Select <0>	0

Table 32 -- Test Pulse Charge Pump Reference Current: Page 1 Address 00010000

Bit	Function	Default=5mA
7	Test Pulse Charge Pump Reference Current (MSB)	0
6	Test Pulse Charge Pump Reference Current	1
5	Test Pulse Charge Pump Reference Current	1
4	Test Pulse Charge Pump Reference Current	0
3	Test Pulse Charge Pump Reference Current	0
2	Test Pulse Charge Pump Reference Current	1
1	Test Pulse Charge Pump Reference Current	0
0	Test Pulse Charge Pump Reference Current (LSB)	0

Note: Current = 5mA default. Full range = 12.7mA (see bias generator section).

Table 33 -- Test Pulse Charge Pump Mirror Cascode Voltage: Page 1 Address 00010001

Bit	Function	Default=400mV
7	Test Pulse Charge Pump Mirror Cascode Voltage (MSB)	0
6	Test Pulse Charge Pump Mirror Cascode Voltage	0
5	Test Pulse Charge Pump Mirror Cascode Voltage	1
4	Test Pulse Charge Pump Mirror Cascode Voltage	1
3	Test Pulse Charge Pump Mirror Cascode Voltage	1
2	Test Pulse Charge Pump Mirror Cascode Voltage	1
1	Test Pulse Charge Pump Mirror Cascode Voltage	1
0	Test Pulse Charge Pump Mirror Cascode Voltage (LSB)	1

Note: Default gives 0.4V. Full range = 0.8V (see bias generator section)



**Table 34 -- CWD Window and Coincidence Offset: Page 1 Address 00011000**

Bit	Function	Default
7	CWD_width<1>	0
6	CWD_width<0>	0
5	Offset_2<2>	0
4	Offset_2<1>	0
3	Offset_2<0>	0
2	Offset_1<2>	0
1	Offset_1<1>	0
0	Offset_1<0>	0

Table 35 -- Miscellaneous Stub Logic: Page 1 Address 00011001

Bit	Function	Default
7	Pt_width<3>	0
6	Pt_width<2>	0
5	Pt_width<1>	1
4	Pt_width<0>	1
3	Fast OR Output Choice (0=Not Latched, 1=Latched)	0
2	Fast OR enable	0
1	Fast OR Source Select (0=OR 254, 1=OR Stubs)	0
0	SR CLKB Enable (0=Shift Clock Dissabled)	0

Table 36 -- Mask Channel<1:8>: Page 1 Address 00100000 (repeats up to <241:248> Address 00111110)

Bit	Function	Default
7	Mask Channel<8>	0
6	Mask Channel<7>	0
5	Mask Channel<6>	0
4	Mask Channel<5>	0
3	Mask Channel<4>	0
2	Mask Channel<3>	0
1	Mask Channel<2>	0
0	Mask Channel<1>	0

Note: In the default case all channels are masked

**Table 37 -- Mask Channel<249:256>: Page 1 Address 00111111**

Bit	Function	Default
7	Unused	0
6	Unused	0
5	Mask Channel<254>	0
4	Mask Channel<253>	0
3	Mask Channel<252>	0
2	Mask Channel<251>	0
1	Mask Channel<250>	0
0	Mask Channel<249>	0

Table 38 -- Channel Offset Word<1>: Page 2 Address 00000001 (repeats up to <253> Address 11111101)

Bit	Function	Default
7	Channel Offset <7>	1
6	Channel Offset <6>	0
5	Channel Offset <5>	0
4	Channel Offset <4>	0
3	Channel Offset <3>	0
2	Channel Offset <2>	0
1	Channel Offset <1>	0
0	Channel Offset <0>	0

Table 39 -- Channel Offset Word<254>: Page 2 Address 11111110

Bit	Function	Default
7	Channel Offset <7>	1
6	Channel Offset <6>	0
5	Channel Offset <5>	0
4	Channel Offset <4>	0
3	Channel Offset <3>	0
2	Channel Offset <2>	0
1	Channel Offset <1>	0
0	Channel Offset <0>	0



Table 40 -- Channel Offset Dummy Word: Page 2 Address 11111111

Bit	Function	Default
7	Channel Offset Dummy <7>	1
6	Channel Offset Dummy <6>	0
5	Channel Offset Dummy <5>	0
4	Channel Offset Dummy <4>	0
3	Channel Offset Dummy <3>	0
2	Channel Offset Dummy <2>	0
1	Channel Offset Dummy <1>	0
0	Channel Offset Dummy <0>	0



CBC2 Channel Test Input Method

Author: M.L. Prydderch / P. Murray

Issue: 1.0.

Published: 28/05/12

NOTE: This document is based on an original document by Peter Murray entitled “CBC2 chip calibration system”. The document was revised to improve detail.



Specification

The CBC2 analogue test input circuit consists of a programmable delay line (regulated by a delay locked loop), coupled to a charge step generator. The delay line allows the time at which the test signal is applied to the analogue inputs to be adjusted by small increments within the time frame of one master clock cycle. The step generator uses a programmable DAC to adjust the amplitude of the test signal. The system has the following specification:

- Delay resolution: 1ns steps @ 40MHz Master Clock.
- Delay range: 25ns @ 40MHz Master Clock.
- Delay variation: +/- 100ps in 25ns or < +/- 5ps in 1ns. NOTE: The nature of the circuit is such that the variation will increase with lower clock frequencies.
- Charge step resolution: 0.04fC
- Charge step dynamic range: 10fC (ie 8 bit)
- Charge step variance due to calibration capacitor variation: 0.004fC (+/-10%)
- Step polarity: Positive or negative (pre-programmed)
- Power supply: 1.2V (digital) 1.1V (analogue)

General description

The block diagram of the system is shown in [Figure 26](#). The master clock is input to the 25 element edge delay line in the Delay Locked Loop (DLL) circuit. The delay comparator compares the time of the $n + 1^{\text{th}}$ positive clock edge at the input to the delay line, with the time of the n^{th} clock edge at the output of the delay line. If the two coincide perfectly then the delay through the line is exactly equal to the clock period and each element of the delay line is therefore a known fraction of the clock cycle period. If the delayed edge arrives too early or too late, the propagation speed is wrong and the state of the delay comparator tells the bias generator to adjust the bias (nbias and pbias) to the starved inverters that make up the delay line elements. The value of these biases sets the speed of the starved inverters and hence the delay through the delay line. The biases are adjusted until the two edges exactly coincide.

The DLL requires 40 to 50 clock cycles to lock to the master clock at start up. Once lock is established the test pulse request signal can then be sent at any convenient time. This signal is a digital, active high signal at least 1 clock cycle long and synchronized to the master clock at the input to the chip. It is input to the shadow delay line which is a copy of the one in the delay locked loop, controlled by the same bias voltages, so the delay through the shadow delay line matches that of the DLL. The timing of the calibration pulse is determined by selecting a tap off point along this shadow delay line, using a 25 to 1 multiplexer. This multiplexer is controlled using the 5 "Edge delay" bits loaded into a register by the I²C interface.



Fig 1

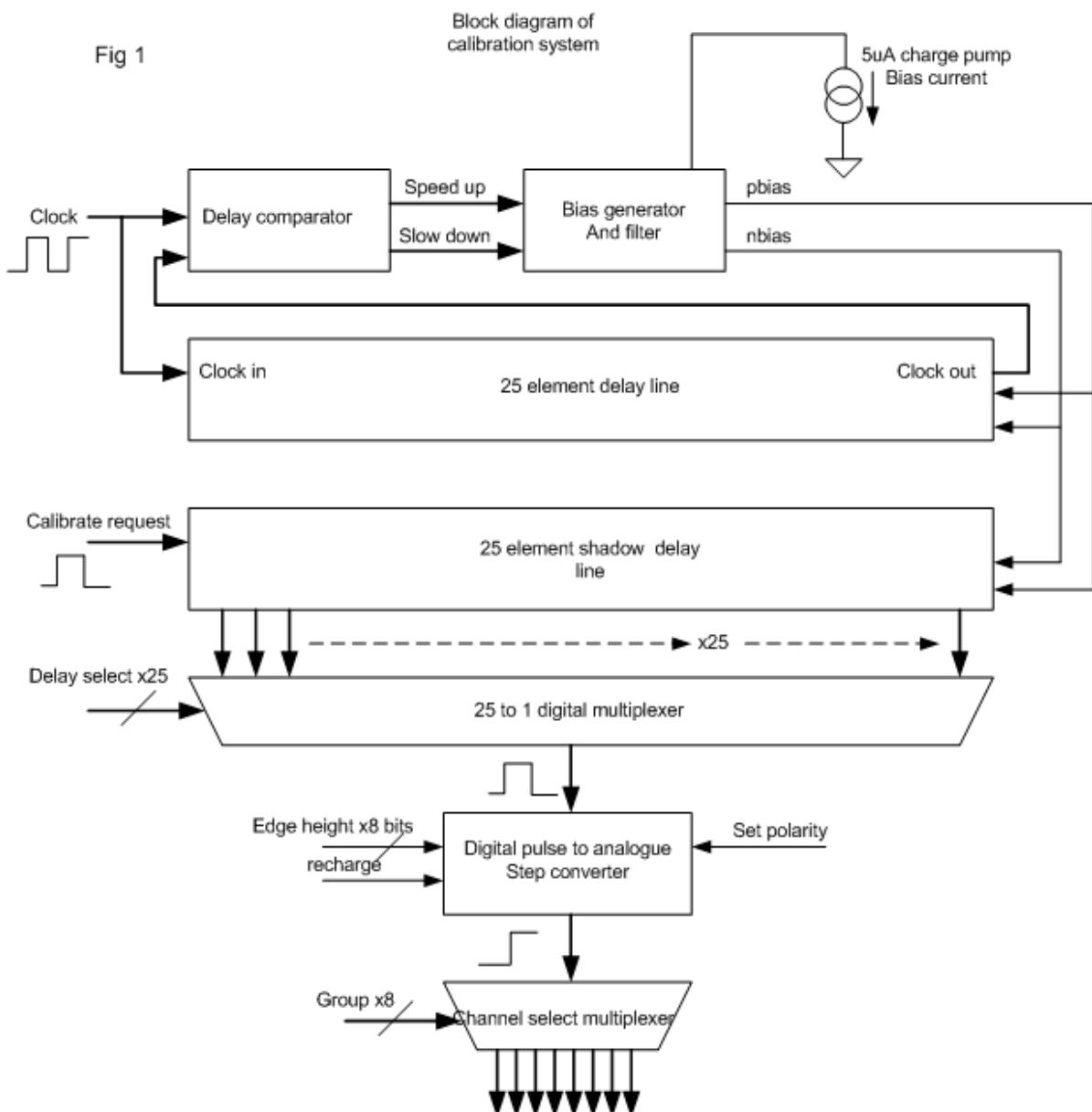


Figure 26.

The 25:1 multiplexer connects the delay tap off point to the “digital pulse to analogue step” converter, the output of which is applied to the test input capacitors on the input to the analogue channels via another stage of multiplexing. The test input capacitors are connected in groups of 32, spaced such that the nearest capacitor of the same group is separated by 7 channels. This last stage of multiplexing determines to which of the 8 groups of 32 capacitors the test pulse is applied. The pulse height is definable with 8 bit accuracy and can be of either polarity. The 8 bits determining pulse height, plus the one determining polarity, are loaded into a register via the I²C interface. The group to be calibrated is selected by a 3 bit word written via the I²C. The user can choose to ground the calibration inputs of all the calibration capacitors not being calibrated, by setting a register bit again



loaded via the I²C. Following a test pulse the step generator node must be recharged so that another calibration step can be generated as required. The recharge action is controlled automatically by a circuit that produces a recharge control signal a 4 clock cycles after the Test Input Request signal is raised.

Voltage Step Generator.

Refer to [Figure 27](#). The desired step size is determined by tapping off a point on the resistor string potentiometer. The 8-bit ‘Step Size’ word is decoded to decide which of the tap off switches to close (s1 in [Figure 27](#)). The switches s3 and s4 are normally open, allowing node A to be charged to the selected potential. The group select decoder decides which group of 32 channels is to receive the test pulse by closing the appropriate switch (s5). The set up of the Step Size and Channel Group is controlled by loading a register via the I2C and must be done before the test pulse is applied.

When the Test Pulse Request command is received by the chip, the pulse is input to the programmable delay line. The delayed pulse out of the delay line triggers the closing of switches s3 or s4 depending on the value of the programmed polarity bit. When the appropriate switch closes, node A ‘discharges’ to 1.1V or 0V. As node A is connected to the calibration capacitors of the selected channels, a voltage step is applied to one plate of these capacitors, thereby producing the charge injection. The series resistor R limits any leakage current out of the potentiometer that might go through s3 or s4 and thus disturb the final voltage attained.

NOTE: Some details have been left out of [Figure 27](#). for clarity. This includes the optional grounding of unused calibration capacitor nodes.

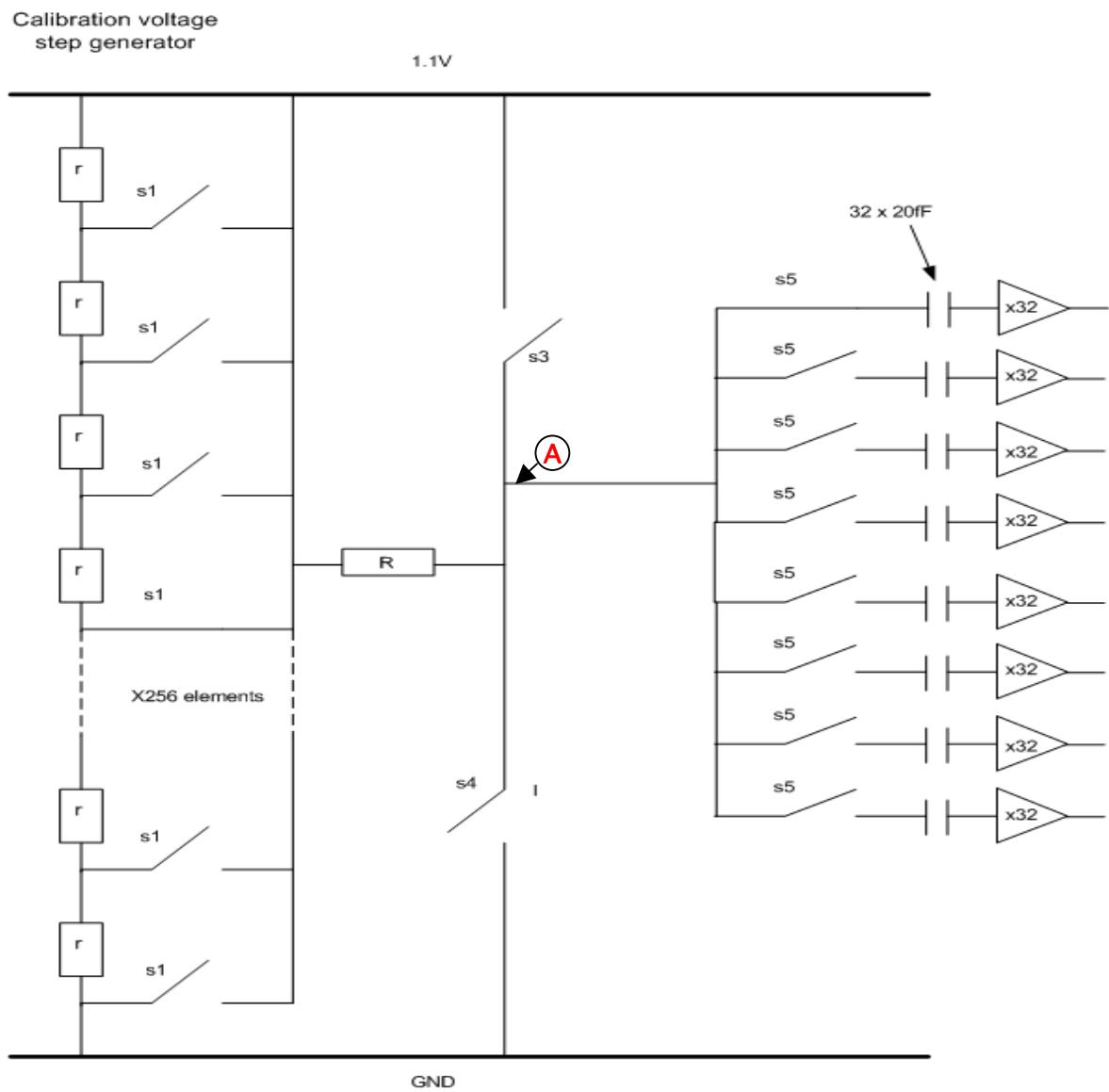


Figure 27.

Test Sequence

Figure 28 shows a simulation of the operation sequence for the injection of a test pulse. The required delay, step value and polarity, the channel group and whether to ground unused calibrate inputs have already been programmed via the I²C.

The details of how to store these values and their locations within the chip memory are to be found in the “CBC2 Register Map” application note.



The green trace in the middle of [Figure 28](#) shows the external TEST_PULSE_REQ signal which triggers the generation of a test signal. The magenta trace above this shows the synchronized version of the same signal internal to the chip. The green trace to the bottom of the figure shows a negative analogue voltage step occurring as a result of the TEST_PULSE_REQ input. The cyan trace at the top of the figure shows the recharge control pulse generated on chip, 4 clock cycles after the TEST_PULSE_REQ signal is received. This signal opens the s3 and s4 switches and allows node **A** to recharge to the selected voltage value ready for another test pulse request. The recharge action of node **A** can be better seen in [Figure 29](#). Figures 5 and 6 show the same simulation for the opposite polarity.

NOTE: The time needed to recharge the step generator to within 8-bit resolution is about 10us.

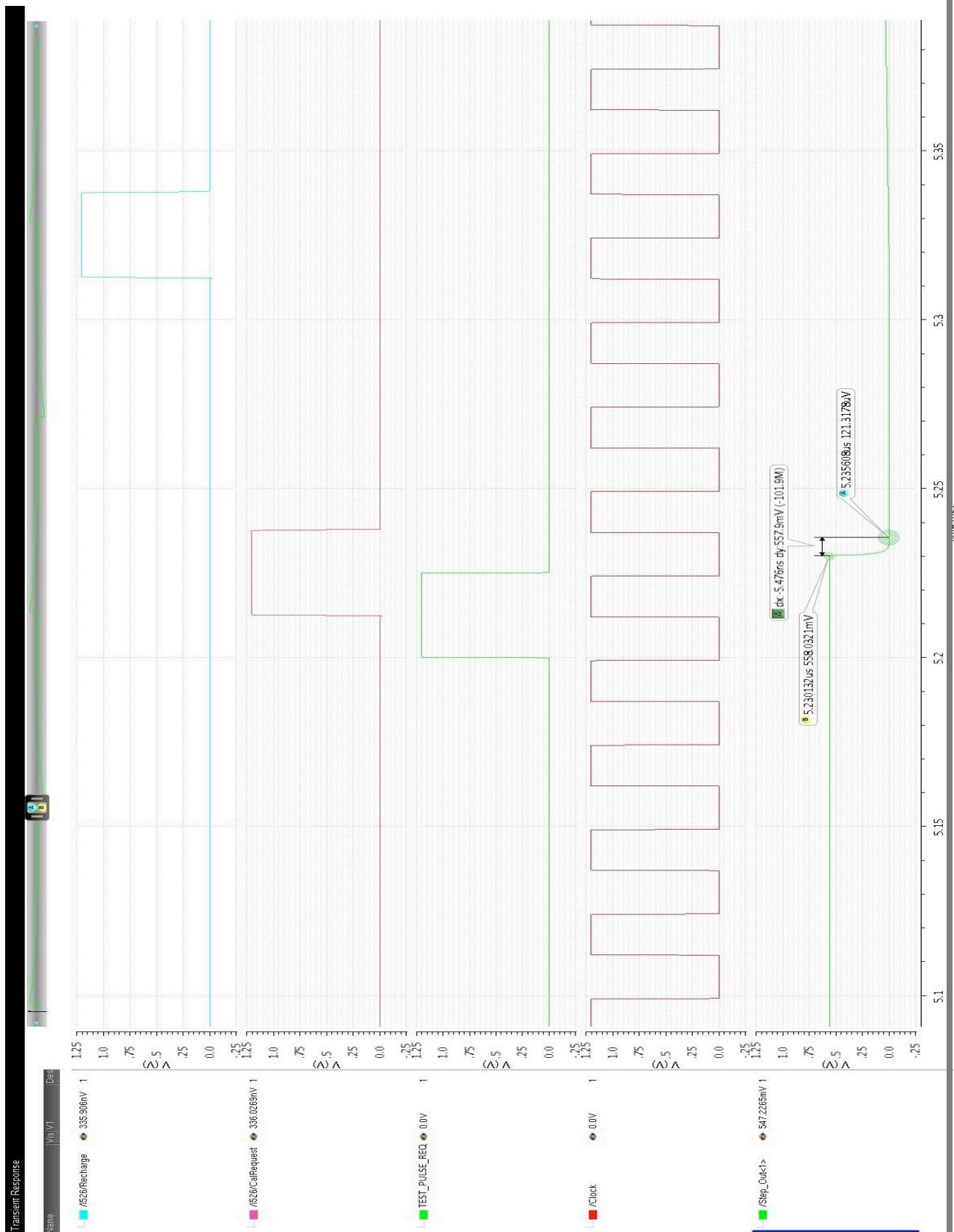


Figure 28.
Science & Technology Facilities Council
Technology

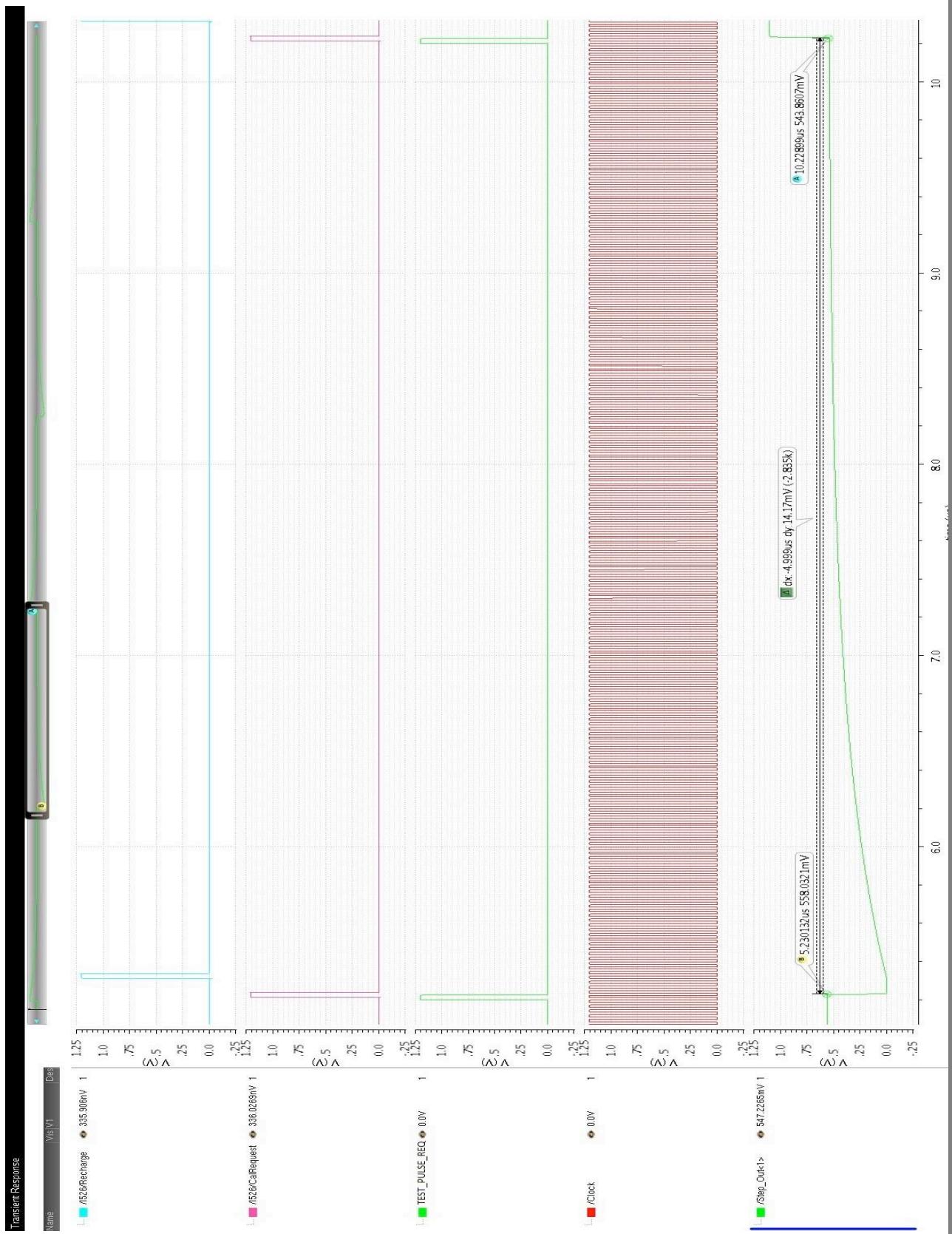


Figure 29.

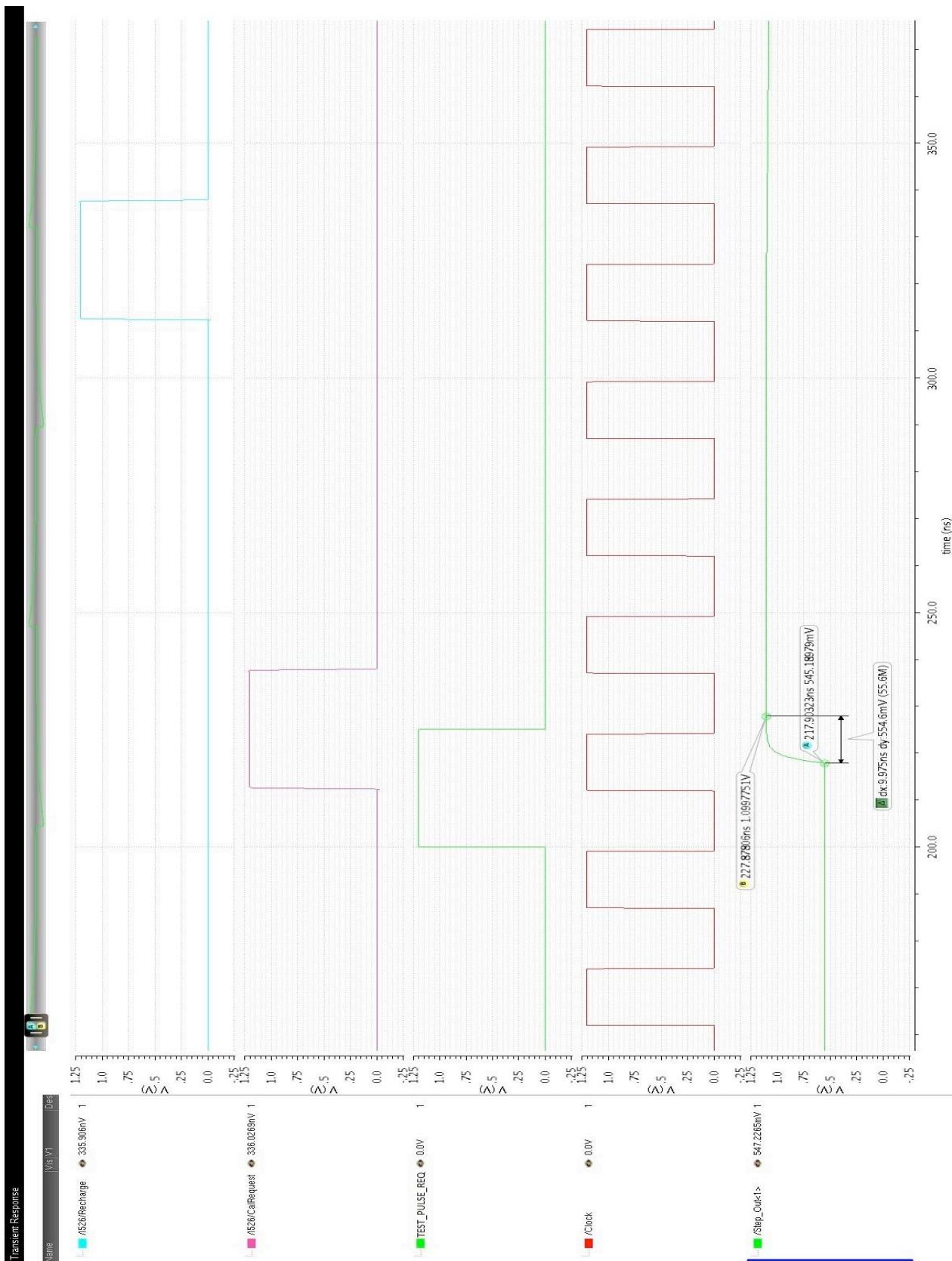


Figure 30.

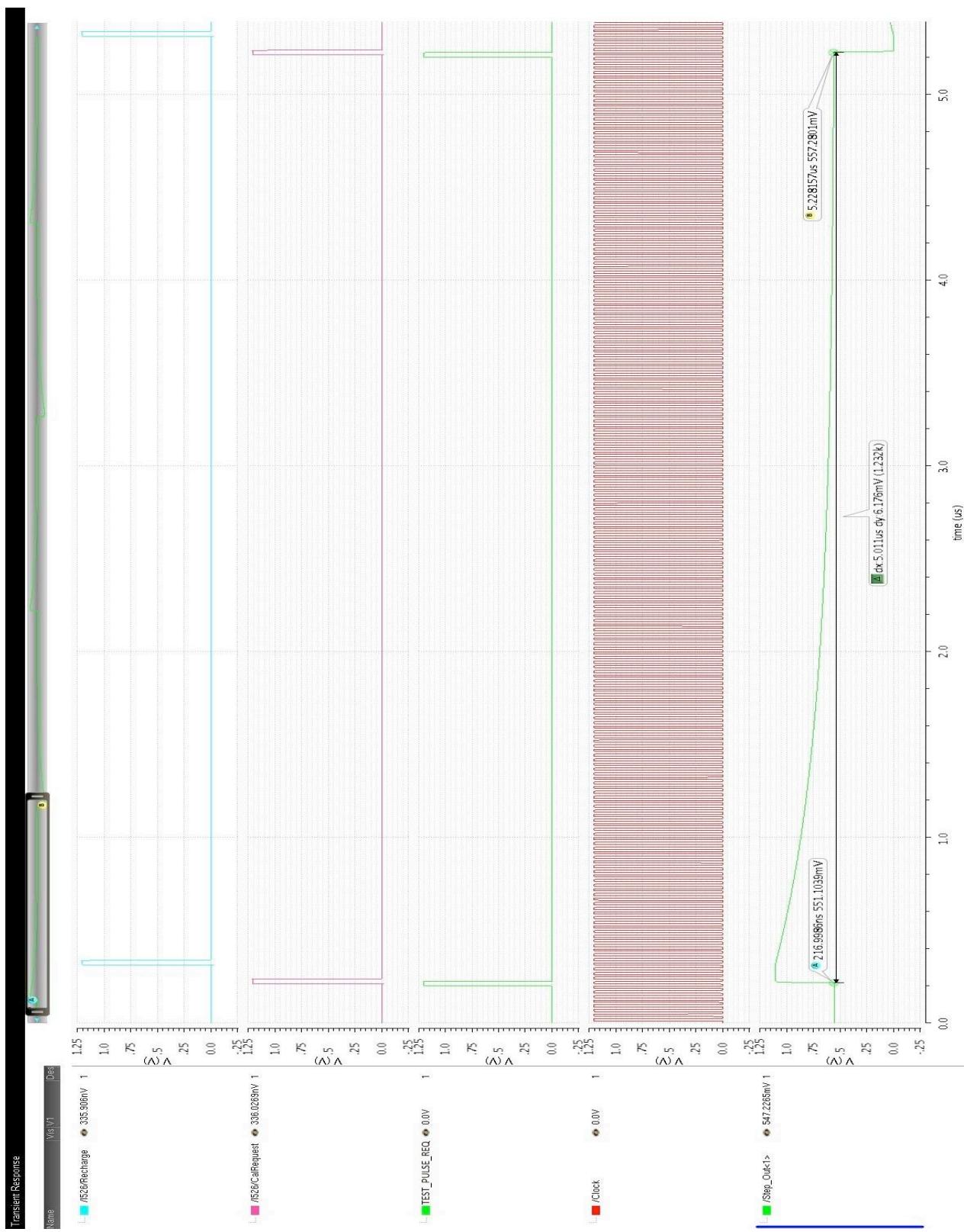


Figure 31.