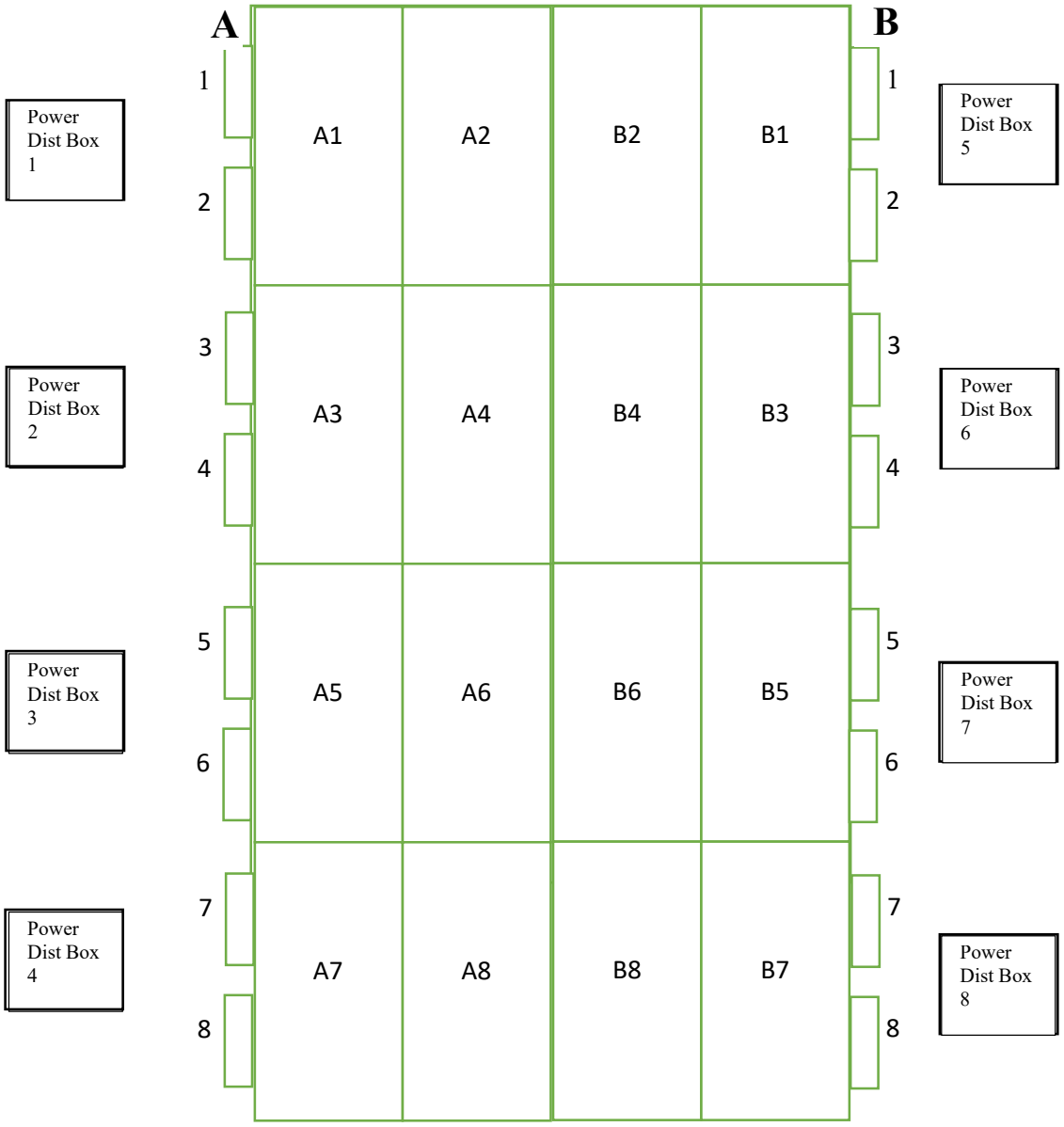
HCal LED pulser system

Overview

Sixteen pulser boxes are mounted on the sides of HCal. Each contains a pulser-board with six LEDs which are each viewed by 18 optical fibers used to distribute 6 different flash brightnesses to 18 PMTs in a 3-wide by 6-high set of modules.



View of HCal from behind. Detector is 12 modules wide by 24 modules high. Each region, served by a single pulser box, is 3 modules wide by 6 modules high. Each pair of pulser boxes is controlled by a power-distribution box, as shown schematically. Power-distribution boxes are actually located on the sides of the electronic racks on the gantry behind HCal.

The six LEDs in each box are numbered 1 through 6, where LED n is approximately twice as bright as LED (n-1). By flashing a set of LEDs in a pulser box simultaneously, the total amount of light delivered to the 18 PMTs serviced by that box can be chosen on a scale of (approximately) 0 through 63 times the brightness of LED 1. The simultaneous flashes of the selected blue LEDs in a pulser box have durations of approximately 20 ns, intended to simulate the response of the scintillators, wavelength shifters, and light guides to incident hadrons.

Eight power-distribution boxes are mounted on the sides of the electronics racks on the gantry. Each power-distribution box contains two power-distribution boards, allowing it to control two pulser boxes. Red LEDs on each power-distribution board indicate which of the six LEDs will fire in response to the next 'Pulse' signal. (The power-distribution boards supply 160V to the selected capacitors which will discharge through their corresponding blue LEDs when the common 'Pulse' signal is received.)

Flashing the Pulsers

A 'Pulse' signal (usually 10 Hz during data-taking) is sent from a frequency-selectable VME board (DVCS pulser) in the electronics bunker to the HCal gantry. (See "How-To" on Prescale and Trigger to see how to set the frequency of this pulser.) There, it is converted to eight TTL signals which are sent to the eight power-distribution boxes. Each signal is then split and sent to the two pulser boxes controlled by each power-distribution box. The 'Pulse' signal triggers avalanche transistors which quickly discharge capacitors across the six blue LEDs. If an LED is desired to not flash, no voltage is applied across the corresponding capacitor so the capacitor provides no charge to power the LED in response to the next Pulse signal. The capacitors have been fine-tuned to store (approx.) twice as much charge in capacitor n as in capacitor (n-1) to give the desired relative brightness of each LED. Applying 160 V across an LED causes it to flash very brightly and would be fatal to the LED within microseconds. A few nano-Henry inductor is connected across each LED to protect it. The inductor presents a high impedance at first, allowing the LED to flash brightly (a 20 ns flash is visible with the naked eye!) but quickly acts like a short, terminating the high current through the LED and determining the duration of the flash.

Normally the 'Pulse' signal comes many times for each selected brightness before the brightness is re-programmed. After each pulse the capacitors recharge (relatively slowly because of high resistances chosen to limit currents) over many micro-seconds. Once the capacitors have recharged, the pulser boards are ready for another 'Pulse' signal and will flash the LEDs with the same brightness. The reproducibility of the brightness the LEDs allows it to be quantified as the variations in brightness result mainly from Poisson statistical fluctuation of the number of photo-electrons. Thus, the mean number of photo-electrons in a flash is:

$$N = \left(\frac{N}{\sqrt{N}}\right)^2 = \left(\frac{B}{\sigma}\right)^2$$

where B is the mean brightness of the flash (proportional to N) and σ is the width of the observed B distribution (proportional to $\sqrt{N}$) with the same proportionality constant, which cancels.

Structure of Power-Distribution Boxes

Each power-distribution box contains two power-distribution boards and a 'connections' board. The two power-distribution boards each control one of the pulser boxes attached to the box. The connection board serves two purposes. First, it secures the many wires coming into and out of the box and going to the boards (clumps of same-colored wires are soldered together under the board). Second, it contains a line-driver chip which takes TTL signals coming into the box (CLK and Data, see below) and increases the current in the signals to allow them to drive the cables leading to the next box in series.

The power-distribution boxes have 4 input connectors (in addition to the Pulse signal which they passively split) and 2 output connectors. The inputs are DC power (+5V on the center of a connector of a BNC cable and ground on the shield), pulser-power (+160 V used to charge the capacitors of the LEDs which are desired to flash, supplied by SHV cables), CLK and Data, see below. The two outputs are copies of CLK and Data made by the line-driver chip on the connection board. In addition a DB25 connector supplies each pulser board with the individual pulser-power lines for each of the 6 LEDs as well as a ground wire.

For safety, the SHV feed-through is immediately connected to a 12 kΩ resistor to limit the maximum current to well below the no-let-go threshold.

The power-distribution boards contain shift-registers which hold the 6-bit binary code for the desired pattern of which blue LEDs should fire on the corresponding pulser board in response to subsequent Pulse signals. The power-distribution boards also carry red LEDs whose only purpose is to allow the pattern in the corresponding shift-register to be read. The bits of the shift register each control the FET-based 'switch' which decides whether a corresponding blue LED has the 160 V power supplied to it.

The 8-bit switch registers on the power-distribution boards allow some manual control. Switches 1-6 can be used to manually disable the output of the corresponding bit, even if it is set in shift-register. The red LED for the disabled bit will not light up and the FET-based switch will not send 160 V power to the disabled bit, even if the corresponding shift-register bit is a 1. For the GMn run, switch 6 was turned off, limiting the maximum brightness going to the PMTs to 31 instead of 63. The 6^{th} bit (brightness 32 time the 1^{st} bit) was brighter than needed. This was included for future uses with higher energies, or if fibers or blue LEDs degrade.

Another use of the switch registers is purely manual operation without a DAQ system. If only certain switches are left on and a set of 96 (or more) 1's is clocked in, then the values indicated by the red LEDs on the power-distribution boxes and the pattern reflected in which capacitors charge up to power the blue LEDs in the pulser boxes, will both be determined by the pattern manually set by the choice of which switches are on. Clocking-in many 1's is easy because a disconnected TTL input acts as a 1. By un-terminating the Data input of the first power-distribution box, the input to the shift register chain is effectively set to 1. If any TTL pulse sequence (up to at least a few kHz frequency, such as the usual Pulse signal) is plugged into the CLK input of that box, in place of the actual CLK signal, then a train of 1's will be clocked in, filling the shift registers with 1's (in a few milli-seconds to a few seconds, depending on the input frequency. The LED pattern will then be decided by which of switches 1-6 are on.

The 7^{th} bit on the switch register does nothing. The 8^{th} bit was designed for use of the board in a light-tight box along with a PMT being tested. It has on effect on the choice of which blue LEDs receive +160 V to charge up their capacitors, but it prevents all red LEDs from lighting up.

The color code on the wires in the power-distribution boxes is:

Red=+5V, Black=Ground, Purple=+160V, Green=CLK, Blue=Data IN, White=Data OUT. See below for the meaning of CLK and Data.

Programming the Power-Distribution Boxes

The set of 16 shift-registers (two per power-distribution box) are arranged in series to form one effective shift-register of 6 * 16 = 96 bits. This shift-register is controlled by two TTL signals (CLK and Data) which are produced by a level-converter in the bunker from the ECL output bits (0 and 1, respectively) of the TI in the lower HCal VME crate. The read-out list hcal1_list.c on intelsbshcal1 controls those bits by using a routine in /home/sbs-onl/linuxvme/hcal_pulser/hcalLib.o called hcalClkIn(iset). This "clocks-in" to the first shift register, the 6-bit binary representation of iset, where $0 \leq iset \leq 63$. The least-significant bit (LSB) is clocked in first and so moves

farthest into the shift register. Here, "clocking-in" a pattern means setting the Data line to 1 or 0, representing the first bit to be clocked in, then setting the CLK line to 1 while the Data line is held constant. (The shift-register shifts on the rising-edge of CLK, so the time at which CLK drops is not critical.) Then the same sequence is repeated for bits 2, 3, …6. (In case of problems in clocking-in the patterns to the first box, one can change the hcalClkIn to hold the data stable longer before/after the leading-edge of the CLK pulse.)

When 12 new bit values are clocked-in to Box 1, the former contents of Box 1 are clocked into Box 2, and so on. The former contents of Box 8 disappear because the output CLK and Data feed-throughs are not connected to anything. In future, these could be connected to a triggered latch, which would make it possible to check whether the transmission of the bit patterns was error-free.

An important detail is that the Pulse signal must be disabled while new values are being shifted into the Boxes. If it pulsed during sequencing, a very incorrect brightness might result because the bits would not yet be in their intended positions. Furthermore, the RF pulse, which is produced when the blue LEDs fire, might cause errors in the shifting-in of the desired bit pattern. Disabling the Pulse signal is accomplished using output bit 2 of the same TI to form an enable signal. Whenever data are being shifted into Box 1 (and on through the effective 96-bit shift register), the hcalClkIn routine also sets this enable bit to zero. This is not routinely re-enabled after each sequence of 6, 12 or 96 shifts because the actual number of shifts may change depending on MODE of operation. To allow hcal1_list.o to re-enable the Pulse signal, when appropriate, a particular value of the iset argument (outside the allowed range of 0 – 63) is used as a flag to indicate the special operation of re-enabling the Pulse signal. This value was chosen as 69, to commemorate the year of the moon landing.

Programming Modes

The file /home/sbs-onl/rol/hcal.flags decides how and when new values will be clocked into the power-distribution boxes. As shown in the example below, the file can set a value of LED_pulser_MODE. The three valid modes, 0, 1, or 2, are briefly explained in the example and will be explained in more detail below.

```
; Setup HCal Pulser (used only if prescale > -1)
LED_pulser_MODE=1,  ; 0-> clk next value to box 1 all others just move down
; 1-> pulse each octant nstep times with pattern= pulser_step (normal)
; 2-> pulse all octants nstep times with pattern= pulser_step (pulser-only runs)
pulser_nsteps=5,
pulser_step0=3,pulser_nstep0=600,
pulser_step1=6,pulser_nstep1=600,
pulser_step2=9,pulser_nstep2=600,
pulser_step3=12,pulser_nstep3=600,
pulser_step4=15,pulser_nstep4=600,
```

The example above defines 5 "steps", stepN, with N numbered 0 through 4, and 5 values of nstepN, with corresponding numbering. The value of stepN determine the brightness of the new sequence of pulses (it is the value whose binary code will determine which blue LEDs flash for subsequent Pulse signals). The value of nstepN determines the number of times the LEDs will flash at each brightness before advancing the sequence. The value of LED_pulser_MODE determines what is done for each advance of the sequence, as described below.

The simplest MODE, and the one which should be used during data-taking is LED_pulser_MODE=1. This tells hcal1_list.o to advance the value stepN sequentially through all 8 boxes, one at a time. For the the sequence above, Prestart would load zero into all 16 power-distribution boards and then load pulser_step0=3 into each of the first two boards of power-distribution box 1. The 12 blue LEDs associated with that box would flash at brightness 3 (meaning LEDs 1 and 2 would flash) each time the Pulse signal arrives until nstep0=600 pulser triggers have been seen by hcal1_list.o. Then hcal1_list.o would clock in 12 0's to advance the value (pulser_step0=3) into power-distribution box 2 for the next nstep0=600 pulser triggers. This sequence continues until the nstep0=600 pulser

triggers have been accepted with pulser_step0=3 loaded into box 8 (and all others being set to zero so they don't flash). The next line of the example would similarly cause pulser_step1=6 to be clocked into power-distribution box 1 (all others being zero) for nstep1=600 pulser events accepted. That value would advance to box 2 through 8. And so on until pulser_step4=15 has sequenced through all 8 boxes for nstep4=600 pulser event accepted for each box. (Since the binary code for 15 is 001111, these events would fire blue LEDs 1, 2, 3, and 4 until 600 events are accepted for each box). Then the sequence repeats, starting with pulser_step0=3 into each of the first two boards of power-distribution box 1. And so on until the end of the run, when all 8 boxes would be cleared. For simplicity, the same value is always clocked into both boards in a single box. The software in hcal1_list.c could be easily adapted to address individual boards independently, if this were an advantage.

MODE 2 clocks the same value into ALL boxes simultaneously, rather than clocking the value into one box at a time with the others being set to zero. In the above example, if LED_pulser_MODE were set to 2, Prestart would load pulser_step0=3 into all 16 power-distribution boards. That would determine the brightness for all pulses until nstep0=600 pulser triggers have been seen by hcal1_list.o. Then hcal1_list.o would clock in the next value, pulser_step1=6 into all 16 power-distribution boards for nstep1=600 pulser events accepted. And so on until pulser_step4=15 has been loaded into all 16 power-distribution boards for nstep4=600 pulser events accepted. The sequence then repeats (starting by clocking pulser_step0=3 into all 16 power-distribution boards for nstep0=600 pulser events accepted. It would continue repeating the sequence until the end of run. This mode would be faster, and simpler, except MODE=2 should NEVER be used during data taking, unless the following problem is solved!!! For reasons which are not understood… and which appear to defy logic, it has been observed that multi-pulsing of the enabled boxes will occur if more than one box is enabled (has non-zero values in its shift-registers) at a time. It seems likely that this phenomenon is somehow related to the fact that the brief high-current pulses which fire the blue LEDs also produce a burst of RF radiation. This explains how the pulser boxes can "talk to each other" and trigger each other's avalanche transistors to initiate a Pulse. The mystery lies in the fact that they re-fire some tens of milliseconds after the Pulse signal when none of them should be emitting an RF pulse to trigger the others. This multi-firing is acceptable if nothing but Pulser triggers are being taken. It might corrupt actual HCal data if the power-distribution boxes were operated in this mode At a 10 Hz trigger rate, the periodic re-fires will have died down in the 100 ms before the next Pulse signal.

Finally, MODE=0 duplicates the original interpretation of hcal.flags LED sequences. In MODE=0, for the example sequence, Prestart would load zero into all 16 power-distribution boards and then load pulser_step0=3 into each of the first two boards of power-distribution box 1. The 12 blue LEDs associated with that box would flash at brightness 3 (meaning LEDs 1 and 2 would flash) each time the Pulse signal arrives until nstep0=600 pulser triggers have been seen by hcal1_list.o. (This is identical to MODE=1.). BUT there would be no automated loading of zeros to move the pattern sequentially into the next boxes. The second line would cause pulser_step1=6 to be clocked into power-distribution box 1. The value pulser_step0=3 formerly in box 1 would move to box 2 so both boxes would be firing their blue LEDs simultaneously until nstep1=600 pulser triggers have been accepted and seen by hcal1_list.o. Subsequent lines would result in brightness values being loaded into the 8 boxes as:

3 0 0 0 0 0 0 0
6 3 0 0 0 0 0 0 Discussed above
9 6 3 0 0 0 0 0
12 9 6 3 0 0 0 0
15 12 9 6 3 0 0 0
3 15 12 9 6 3 0 0 Cycling through the input lines
6 3 15 12 9 6 3 0
9 6 3 15 12 9 6 3
12 9 6 3 15 12 9 6
15 12 9 6 3 15 12 9 … and so on until the end of the run caused 0 to be loaded in all 12 boxes.

Note that these sequences would cause multi-firing of the boxes because two boxes have non-zero values simultaneously. MODE 0 is flexible but clumsy. Either MODE 1 or MODE 2 could be imitated using MODE 0, but a long sequence of input values of pulser_stepN and nstepN would be needed. Either each non-zero pulser_stepN value would have to be followed by seven pulser_stepN=0 values to emulate MODE 1 or each non-

zero pulser_stepN value would have to be followed by seven identical pulser_stepN values to emulate MODE 2. MODEs 1 and 2 make the more common modes of operation accessible with 8-times shorter sequences in hcal.flags, and greatly reduce the chance of errors.