To: Daniel Blackman, Rick Raffanti
From: G. Murphy
Subj.: Digitizer to DCE interface

After studying the as-built digitizer schematics I would like to put down in memo form a kind of "ICD" for the digitizer command interface. The following statements have been derived from information sent to me by Dan, and from the schematics themselves.

1) Physical interface: There are three signals which go from the DCE to the Digitizer Y board (I'm assuming that there is a separate 3 wire interface to the A and B segments). These signals have the names BSTB, BCLK, and BCMD at the interface to the Y Actel in the current schematic. BSTB is essentially an enable (enable=1), BCLK is a software generated clock (not necessarily a symmetrical clock) that is high for 1.5 microseconds. BCMD is the serial data values (16 bits in all). I can latch the data on EITHER the rising or falling edge of the clock—I will use the rising edge.
2) Word structure: The Data is a sixteen bit word sent MSb first. The structure is as follows

   Msb Lsb

   0xxx 0yyy zzzzzzzz

   Where "xxx" is the "chip select" for the DACs and Mux's, "yyy" is the address for the chip selected, and "zzzzzzzz" is the data value to be latched to the DAC (Non-applicable for the Mux).
3) the Y Actel will clock in the 16 bits of data and on the falling edge of BSTB that data will be latched. On the next clock cycle (4MHz clk), the data will be presented to the DAC's and Mux's accordingly.
4) There are 3 DACs and 3 Mux's. The Most Significant Nibble (MSN) will select chips according to the following table:

Bit # 14*	Bit # 13	Bit # 12	selected
0	0	0	DAC0
0	0	1	DAC1
0	1	0	DAC2
0	1	1	UNDEFINED
1	0	0	MUX0
1	0	1	MUX1
1	1	0	MUX2
1	1	1	STIM ACTIVE

**note: Bit 15 will be ignored, it could be used to validate a command by demanding that it be zero—this decision can be made prior to the flight chip being burned.*

5) DAC#0 handles settings for the X digitizer, DAC#1 handles timing and charge thresholds, and DAC#2 handles settings for the Y digitizer. The "yyy" values select the following values depending on the values of "xxx"

Bit # 10*	Bit # 9	Bit # 8	DAC#0	DAC#1	DAC#2	MUX0	MUX 1	MUX2
0	0	0	X_SHIFT	TTHR_Y	Y_SHIFT -	X_SHIFT	+15	Y_SHIFT
0	0	1	X_STRETCH	Q_LO	Y_STRETCH	X_STRETCH	-15	Y_STRETCH
0	1	0	B_WALK_X	Q_HI	B_WALK_Y	B_WALK_X	Vcc	B_WALK_Y
0	1	1	E_WALK_X	TTHR_X	E_WALK_Y	E_WALK_X	-5.2	E_WALK_Y
1	0	0	-	-	-	TTHR_Y	VREF	GND
1	0	1	-	-	-	Q_LO	GND	GND
1	1	0	-	-	-	Q_HI	GND	GND
1	1	1	-	-	-	TTHR_X	GND	GND

**note: Bit 11 will be ignored, it could be used to validate a command by demanding that it be zero—this decision can be made prior to the flight chip being burned.*

6) Although it doesn't really matter from the Actel design point of view, I will assume that operationally, you first send a command to set a particular DAC value, then issue another command to output that value on the Mux. Once the DAC is set, it remains set, only one mux value can be selected at a time.