

FPGAs and Verilog Lab

Implement a chronograph

Objective

Implement in a FPGA development board a chronograph

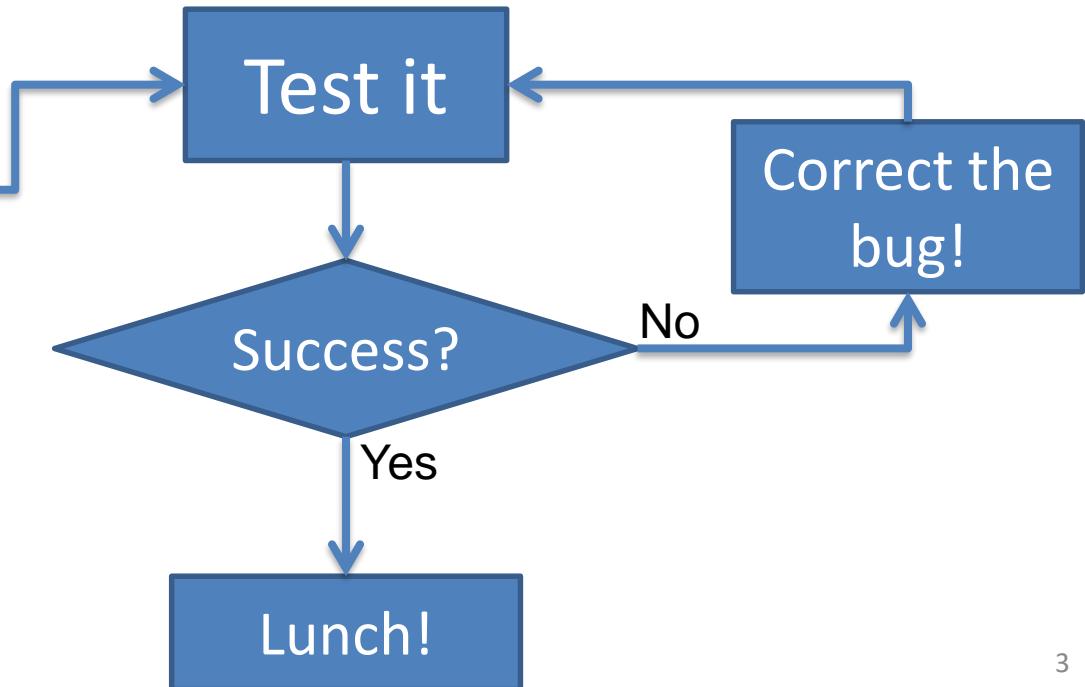
Count seconds from 0 to 99 when a switch is up...

First Step

Open the Quartus Software;
Open the DE2_top project
compile;
program the board

-- The DE2_top gives you the definition of all pins and signals in DE2

Do Something different...
e.g. Change the value of
the 7 Segments display
compile
program



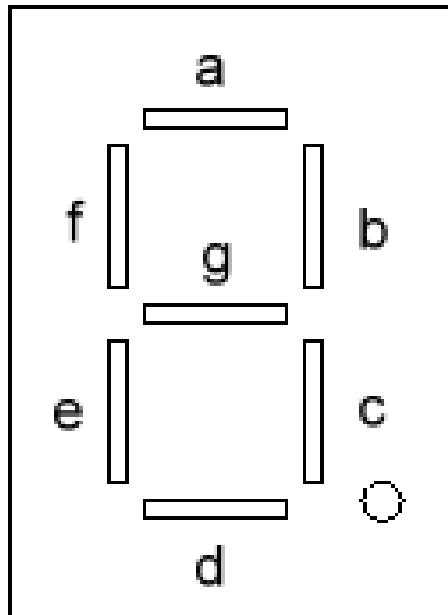
Decode the binary and put it in 7 seg display

Do a module that decodes binary to 7 segment display

Use the Switches as inputs and output to Hex6 and 7

SW[3:0] → Hex6

SW[7:4] → Hex7



```
case (bin)
//disp= "g f e d c b a"
0: display=~7'b0111111;
1: display=~7'b0000110;
2: display=~7'b1011011;
3: display=~7'b1001111;
4: display=~7'b1100110;
5: display=~7'b1101101;
6: display=~7'b1111100;
7: display=~7'b0000111;
8: display=~7'b1111111;
9: display=~7'b1100111;
default:display=~7'b0000000;
endcase
```

Flash a LED with T=1s

Produce a one second clock and change a led each second

Identify the 50 MHz clock.

Divide the clock using a 32 bit counter:

Create a 32 bit counter and show in the LEDs

Now, everytime the counter value reaches $1\text{s}/(\text{clock period}) = 1\text{s}/20\text{ns}$
activate a flag and reset counter

Now you have a pulse each second...

Use it to turn on/off a LED

Count 1 second pulses

Make two counters: seconds and tens of seconds

Use the 1s pulse for the seconds.

Each 10 seconds give a pulse to the tens of seconds and
reset the seconds counter

Control the counter

Use a switch to control when it counts

Use a Key for reset:

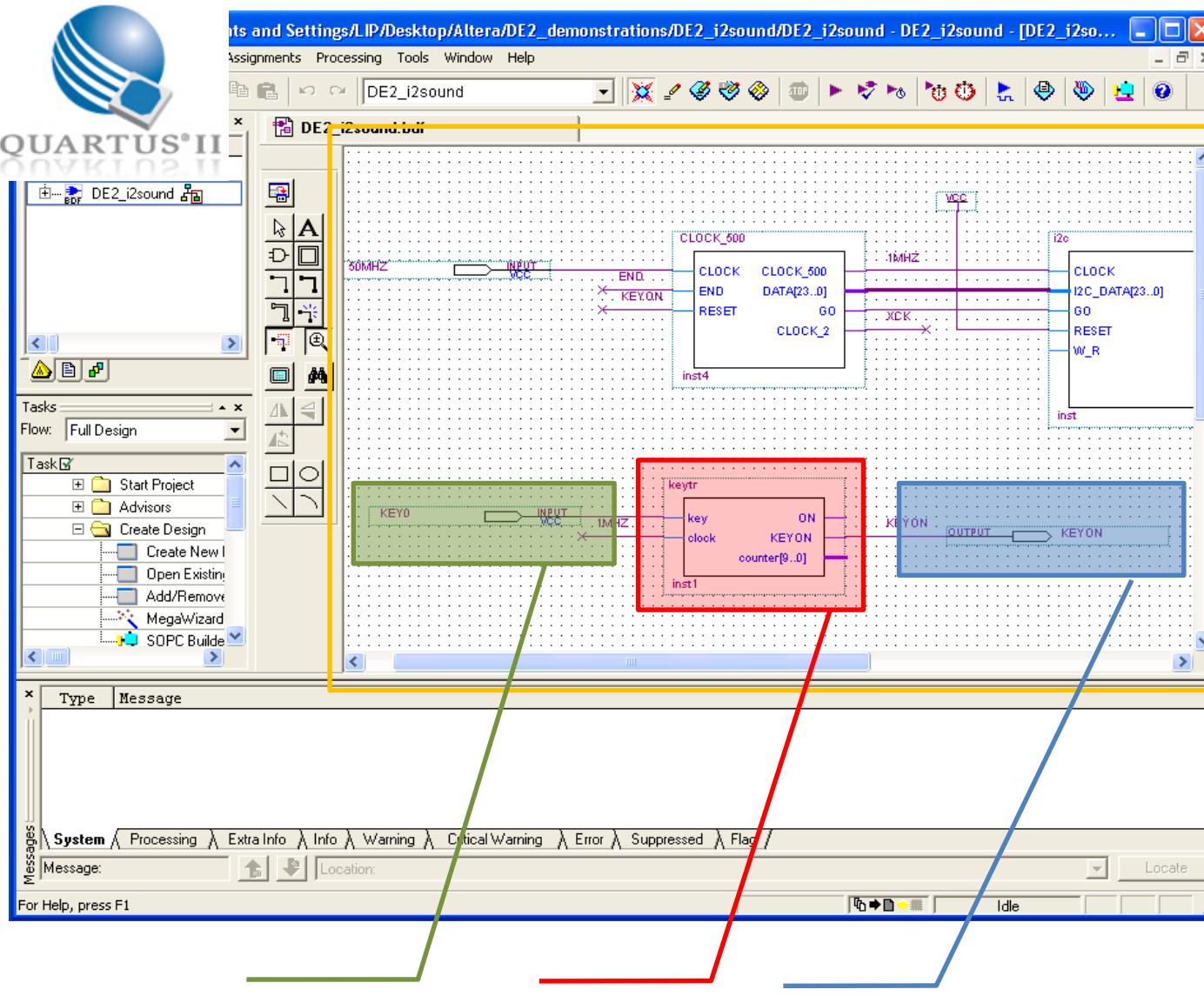
```
wire reset;  
assign reset=Key[0];  
always @ (posedge ... or negedge reset)  
if (!reset)  
    count <=0;  
else .....
```

Have fun with it

Nibble to 7 segment

```
case (nibble)
  0: segment <= ~8'b11111100; // "a b c d e f g h"
  1: segment <= ~8'b01100000;
  2: segment <= ~8'b11011010; // --a--
  3: segment <= ~8'b11110010; // |   |
  4: segment <= ~8'b01100110; // f   b
  5: segment <= ~8'b10110110; // |   |
  6: segment <= ~8'b10111110; // --g--
  7: segment <= ~8'b11100000; // |   |
  8: segment <= ~8'b11111110; // e   c
  9: segment <= ~8'b11100110; // |   |
  10: segment <= ~8'b11101110; // --d-- (h)
  11: segment <= ~8'b00111110;
  12: segment <= ~8'b10011100;
  13: segment <= ~8'b01111010;
  14: segment <= ~8'b10011110;
  15: segment <= ~8'b10001110;
  default: segment <= 8'bx;
endcase
```

Software - Schematic



Input

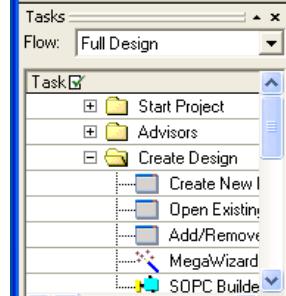
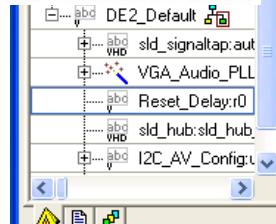
Logic

Output

Schematic
programming



QUARTUS® II



```
1 module Reset_Delay(iCLK,oRESET);
2   input iCLK;
3   output reg oRESET;
4   reg [19:0] Cont;
5
6   always@ (posedge iCLK)
7   begin
8     if(Cont != 20'hFFFFF)
9     begin
10       Cont <= Cont+1;
11       oRESET <= 1'b0;
12     end
13     else
14       oRESET <= 1'b1;
15   end
16
17 endmodule
```

Verilog Programming

Input

Output

Logic

Implementing a Multiplexer

```
module mux(f, a, b, sel);
output f;
input a, b, sel;

and g1(f1, a, nsel),
      g2(f2, b, sel);
or  g3(f, f1, f2);
not g4(nsel, sel);

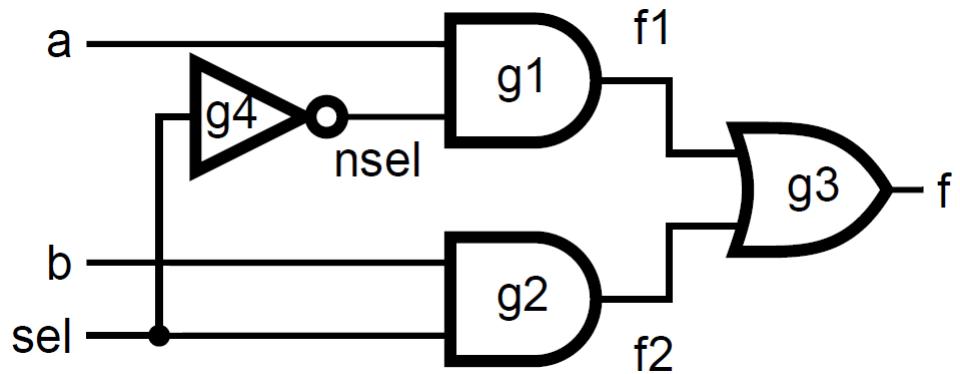
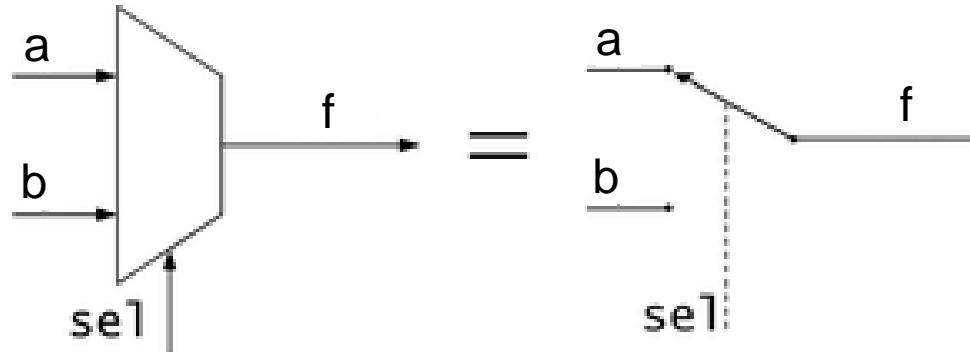
endmodule
```

```
module mux(f, a, b, sel);
output f;
input a, b, sel;

reg f;

always @ (a or b or sel)
  if (sel) f = a;
  else f = b;

endmodule
```



```
module mux(f, a, b, sel);
output f;
input a, b, sel;

assign f = sel ? a : b;

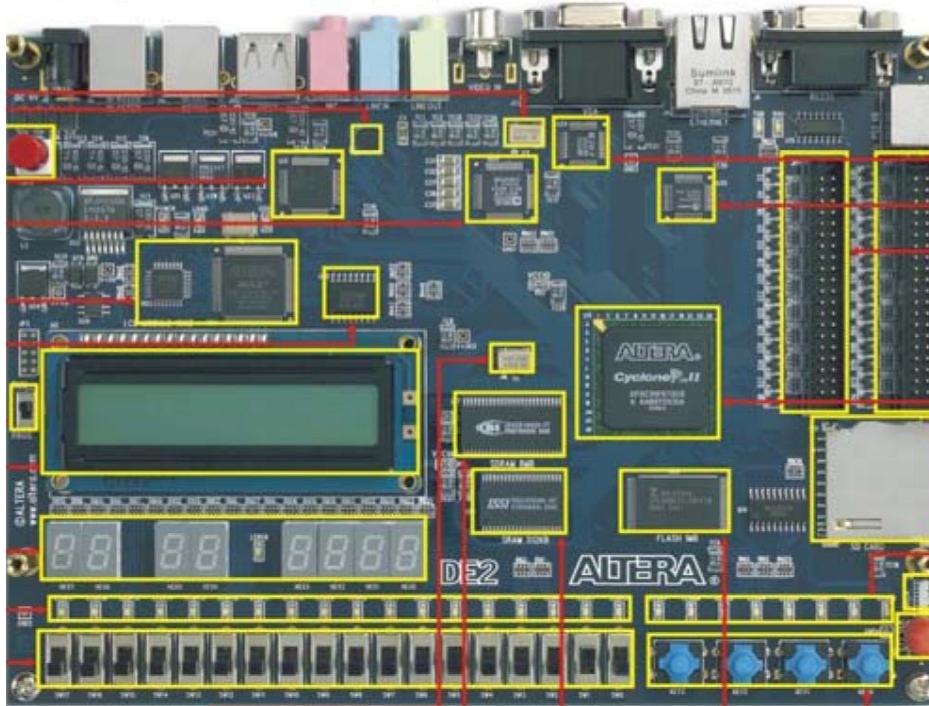
endmodule
```

Hardware: buy and hack it

Hardware



Hardware



Hardware

DE2 Pin Table.pdf

Altera DE2 Board Pin Table

Digital Name	PIN# Pin No.	Driver Name
DIGITAL_N	PIN_N25	Toggle_Swift[0]
SWI[1]	PIN_N25	Toggle_Swift[1]
SWI[2]	PIN_A14	Toggle_Swift[2]
SWI[3]	PIN_AF14	Toggle_Swift[3]
SWI[4]	PIN_A13	Toggle_Swift[4]
SWI[5]	PIN_AC13	Toggle_Swift[5]
SWI[6]	PIN_C13	Toggle_Swift[6]
SWI[7]	PIN_A12	Toggle_Swift[7]
SWI[8]	PIN_A13	Toggle_Swift[8]
SWI[9]	PIN_N1	Toggle_Swift[9]
SWI[10]	PIN_P1	Toggle_Swift[10]
SWI[11]	PIN_P2	Toggle_Swift[11]
SWI[12]	PIN_P3	Toggle_Swift[12]
SWI[13]	PIN_U9	Toggle_Swift[13]
SWI[14]	PIN_U4	Toggle_Swift[14]
SWI[15]	PIN_V1	Toggle_Swift[15]
SWI[16]	PIN_V2	Toggle_Swift[16]
SWI[17]	PIN_V3	Toggle_Swift[17]
Digital Name	PIN# Pin No.	Driver Name
DRAM_ADR[0]	PIN_V4	SDRAM_Address[0]
DRAM_ADR[1]	PIN_V5	SDRAM_Address[1]
DRAM_ADR[2]	PIN_V6	SDRAM_Address[2]
DRAM_ADR[3]	PIN_W1	SDRAM_Address[3]
DRAM_ADR[4]	PIN_U8	SDRAM_Address[4]
DRAM_ADR[5]	PIN_U9	SDRAM_Address[5]
DRAM_ADR[6]	PIN_U5	SDRAM_Address[6]
DRAM_ADR[7]	PIN_P8	SDRAM_Address[7]
DRAM_ADR[8]	PIN_W2	SDRAM_Address[8]
DRAM_ADR[9]	PIN_Y1	SDRAM_Address[9]
DRAM_ADR[10]	PIN_VR	SDRAM_Address[10]
DRAM_DQ[0]	PIN_VR	SDRAM_Data[0]
DRAM_DQ[1]	PIN_AA2	SDRAM_Data[1]
DRAM_DQ[2]	PIN_VT	SDRAM_Data[2]
DRAM_DQ[3]	PIN_VS	SDRAM_Data[3]
DRAM_DQ[4]	PIN_V4	SDRAM_Data[4]
DRAM_DQ[5]	PIN_V5	SDRAM_Data[5]
DRAM_DQ[6]	PIN_V6	SDRAM_Data[6]
DRAM_DQ[7]	PIN_T8	SDRAM_Data[7]
DRAM_DQ[8]	PIN_V7	SDRAM_Data[8]
DRAM_DQ[9]	PIN_V8	SDRAM_Data[9]
DRAM_DQ[10]	PIN_AB5	SDRAM_Data[10]
DRAM_DQ[11]	PIN_AB4	SDRAM_Data[11]
DRAM_DQ[12]	PIN_AA3	SDRAM_Data[12]
DRAM_DQ[13]	PIN_AB2	SDRAM_Data[13]
DRAM_DQ[14]	PIN_AC1	SDRAM_Data[14]
DRAM_DQ[15]	PIN_AA5	SDRAM_Data[15]
DRAM_DQ[16]	PIN_AB6	SDRAM_Data[16]
DRAM_BA	PIN_AE3	SDRAM_Bank_Address[0]
DRAM_LDM	PIN_AE2	SDRAM_Low_Direction_Mask
DRAM_RAS	PIN_AB4	SDRAM_Raw_Address_Strobe
DRAM_CAS	PIN_AE3	SDRAM_Col_Address_Strobe
DRAM_RP	PIN_AA7	SDRAM_RP
DRAM_CLK	PIN_AA7	SDRAM_Clock
DRAM_CS	PIN_AC3	SDRAM_CS
Digital Name	PIN# Pin No.	Driver Name
PL_ADDRESS[0]	PIN_AB18	FLASH_Address[0]
PL_ADDRESS[1]	PIN_AE19	FLASH_Address[1]
PL_ADDRESS[2]	PIN_AC19	FLASH_Address[2]
PL_ADDRESS[3]	PIN_AE18	FLASH_Address[3]

Altera DE2 Board Pin Table

Signal Name	PIN#	Function
LED(R)	PIN_A01	LED Red(0)
LED(G)	PIN_A22	LED Green(1)
LED(B)	PIN_A23	LED Blue(2)
LED(W)	PIN_A21	LED Yellow(3)
LED(R)	PIN_A14	LED Red(0)
LED(G)	PIN_Y13	LED Green(1)
LED(B)	PIN_Y14	LED Blue(2)
LED(E1)	PIN_AC14	LED Red(1)
LED(E2)	PIN_AC15	LED Red(1)
LED(E3)	PIN_AC16	LED Red(1)
LED(E4)	PIN_AC17	LED Red(1)
LED(E5)	PIN_AC18	LED Red(1)
LED(E6)	PIN_AC19	LED Red(1)
LED(E7)	PIN_AC20	LED Red(1)
LED(E8)	PIN_AC21	LED Red(1)
LED(E9)	PIN_AC22	LED Red(1)
LED(E10)	PIN_AC23	LED Red(1)
LED(E11)	PIN_AC24	LED Red(1)
LED(E12)	PIN_AC25	LED Red(1)
LED(E13)	PIN_AC26	LED Red(1)
LED(E14)	PIN_AC27	LED Red(1)
LED(E15)	PIN_AC28	LED Red(1)
LED(E16)	PIN_AC29	LED Red(1)
LED(E17)	PIN_AC30	LED Red(1)
LED(E18)	PIN_AC31	LED Red(1)
LED(O1)	PIN_AF01	LED Orange(1)
LED(O2)	PIN_W19	LED Orange(1)
LED(O3)	PIN_W20	LED Orange(1)
LED(O4)	PIN_W18	LED Orange(1)
LED(O5)	PIN_W17	LED Orange(1)
LED(O6)	PIN_W16	LED Orange(1)
LED(O7)	PIN_W15	LED Orange(1)
LED(O8)	PIN_W14	LED Orange(1)
LED(O9)	PIN_W13	LED Orange(1)
LED(O10)	PIN_W12	LED Orange(1)
LED(O11)	PIN_W11	LED Orange(1)
LED(O12)	PIN_W10	LED Orange(1)
LED(O13)	PIN_W9	LED Orange(1)
LED(O14)	PIN_W8	LED Orange(1)
LED(O15)	PIN_W7	LED Orange(1)
LED(O16)	PIN_W6	LED Orange(1)
LED(O17)	PIN_W5	LED Orange(1)
LED(O18)	PIN_W4	LED Orange(1)
LED(O19)	PIN_W3	LED Orange(1)
LED(O20)	PIN_W2	LED Orange(1)
LED(O21)	PIN_W1	LED Orange(1)
LED(O22)	PIN_W0	LED Orange(1)
External Name	PIN#	Description
GCLK_27	PIN_B13	On Board 27 MHz Clock
OC	PIN_B14	On Board 27 MHz Clock
EXT_CLOCK	PIN_P25	External Clock
UART_RXD	PIN_C23	UART Receiver
UART_TXD	PIN_B25	UART Transmitter
PZS_CLK	PIN_D28	PZS Data
PZS_DAT	PIN_C28	PZS Clock
I2C_SCLK	PIN_A6	I2C Data
I2C_SDAT	PIN_A7	I2C Clock
Signal Name	PIN#	Description
TD[DATA0]	PIN_J6	TD Decoder Data[0]
TD[DATA1]	PIN_H7	TD Decoder Data[1]
TD[DATA2]	PIN_H8	TD Decoder Data[2]
TD[DATA3]	PIN_H10	TD Decoder Data[3]
TD[DATA4]	PIN_H11	TD Decoder Data[4]
TD[DATA5]	PIN_H12	TD Decoder Data[5]
TD[DATA6]	PIN_H9	TD Decoder Data[6]
TD[DATA7]	PIN_D7	TD Decoder Data[7]
TD[DATA8]	PIN_D6	TD Decoder Data[8]
TD[DATA9]	PIN_D5	TD Decoder Data[9]
TD[DATA10]	PIN_D4	TD Decoder Data[10]
TD[DATA11]	PIN_D3	TD Decoder Data[11]
TD[DATA12]	PIN_D2	TD Decoder Data[12]
TD[DATA13]	PIN_D1	TD Decoder Data[13]
TD[DATA14]	PIN_D0	TD Decoder Data[14]
TD[HE]	PIN_B8	TD Decoder H, BYTCR
TD[ARST]	PIN_C4	TD Decoder Reset
Signal Name	PIN#	Description
VGA_R[1]	PIN_F10	VGA Red(1)
VGA_G[1]	PIN_G10	VGA Green(1)
VGA_B[1]	PIN_E9	VGA Blue(1)
VGA_R[6]	PIN_C9	VGA Red(6)
VGA_R[5]	PIN_A8	VGA Red(5)
VGA_R[4]	PIN_A7	VGA Red(4)
VGA_R[3]	PIN_H12	VGA Red(3)
VGA_R[2]	PIN_H11	VGA Red(2)
VGA_R[1]	PIN_B10	VGA Red(1)
VGA_G[6]	PIN_B9	VGA Green(6)
VGA_G[5]	PIN_B8	VGA Green(5)
VGA_G[4]	PIN_B7	VGA Green(4)
VGA_G[3]	PIN_H10	VGA Green(3)
VGA_G[2]	PIN_H9	VGA Green(2)
VGA_G[1]	PIN_H8	VGA Green(1)

Altera DE2 Board Pin Table

PL Address[1]	PN#	AF#	FLASH Address[5]
PL ADDRESS[9]	PN#_Y16	FLASH Address[0]	
PL ADDRESS[10]	PN#_Y17	FLASH Address[1]	
PL ADDRESS[11]	PN#_AD17	FLASH Address[2]	
PL ADDRESS[12]	PN#_AC17	FLASH Address[3]	
PL ADDRESS[13]	PN#_AF17	FLASH Address[4]	
PL ADDRESS[14]	PN#_W16	FLASH Address[5]	
PL ADDRESS[15]	PN#_W17	FLASH Address[6]	
PL ADDRESS[16]	PN#_AC16	FLASH Address[7]	
PL ADDRESS[17]	PN#_AC15	FLASH Address[8]	
PL ADDRESS[18]	PN#_AB15	FLASH Address[9]	
PL ADDRESS[19]	PN#_AC14	FLASH Address[10]	
PL ADDRESS[20]	PN#_Y15	FLASH Address[21]	
PL ADDRESS[21]	PN#_Y14	FLASH Address[22]	
PL ADDRESS[22]	PN#_AF14	FLASH Address[23]	
PL ADDRESS[23]	PN#_AC19	FLASH [unit1]	
PL ADDRESS[24]	PN#_AC18	FLASH [unit2]	
PL ADDRESS[25]	PN#_AE10	FLASH [unit3]	
PL ADDRESS[26]	PN#_AB10	FLASH [unit4]	
PL ADDRESS[27]	PN#_AC10	FLASH [unit5]	
PL ADDRESS[28]	PN#_AF21	FLASH [unit6]	
PL ADDRESS[29]	PN#_AE21	FLASH [unit7]	
PL ADDRESS[30]	PN#_AB21	FLASH [unit8]	
PL_DIN	PN#_W17	FLASH Output Enable	
PL_RST	PN#_A18	FLASH Reset	
PL_CS	PN#_A14	FLASH Control	
Signal Name	PN#	PN# No.	Description
SRAM_ADDRESS[0]	PN#_AEE	SRAM Address[0]	
SRAM_ADDRESS[1]	PN#_AEC	SRAM Address[1]	
SRAM_ADDRESS[2]	PN#_AC5	SRAM Address[2]	
SRAM_ADDRESS[3]	PN#_AC8	SRAM Address[3]	
SRAM_ADDRESS[4]	PN#_AC9	SRAM Address[4]	
SRAM_ADDRESS[5]	PN#_AOS	SRAM Address[5]	
SRAM_ADDRESS[6]	PN#_AF5	SRAM Address[6]	
SRAM_ADDRESS[7]	PN#_AOD	SRAM Address[7]	
SRAM_ADDRESS[8]	PN#_V10	SRAM Address[8]	
SRAM_ADDRESS[9]	PN#_V10	SRAM Address[10]	
SRAM_ADDRESS[11]	PN#_V9	SRAM Address[11]	
SRAM_ADDRESS[12]	PN#_WB	SRAM Address[12]	
SRAM_ADDRESS[13]	PN#_WB	SRAM Address[13]	
SRAM_ADDRESS[14]	PN#_W10	SRAM Address[14]	
SRAM_ADDRESS[15]	PN#_W10	SRAM Address[15]	
SRAM_ADDRESS[16]	PN#_ABB	SRAM Address[16]	
SRAM_ADDRESS[17]	PN#_ACB	SRAM Address[17]	
SRAM_ADDRESS[18]	PN#_AEC	SRAM Address[18]	
SRAM_ADDRESS[19]	PN#_AEB	SRAM Address[19]	
SRAM_ADDRESS[20]	PN#_AED	SRAM Address[20]	
SRAM_ADDRESS[21]	PN#_AF9	SRAM Address[21]	
SRAM_ADDRESS[22]	PN#_AF9	SRAM Address[22]	
SRAM_ADDRESS[23]	PN#_AA10	SRAM Address[23]	
SRAM_ADDRESS[24]	PN#_AB10	SRAM Address[24]	
SRAM_ADDRESS[25]	PN#_AC11	SRAM Address[25]	
SRAM_ADDRESS[26]	PN#_Y11	SRAM Address[26]	
SRAM_ADDRESS[27]	PN#_Y11	SRAM Address[27]	
SRAM_ADDRESS[28]	PN#_AF7	SRAM Address[28]	
SRAM_ADDRESS[29]	PN#_AF7	SRAM Address[29]	
SRAM_ADDRESS[30]	PN#_AEB	SRAM Address[30]	
SRAM_ADDRESS[31]	PN#_W11	SRAM Address[31]	
SRAM_ADDRESS[32]	PN#_W11	SRAM Address[32]	
SRAM_ADDRESS[33]	PN#_W12	SRAM Address[33]	
SRAM_ADDRESS[34]	PN#_W12	SRAM Address[34]	
SRAM_ADDRESS[35]	PN#_AC15	SRAM Address[35]	

DE2 U4

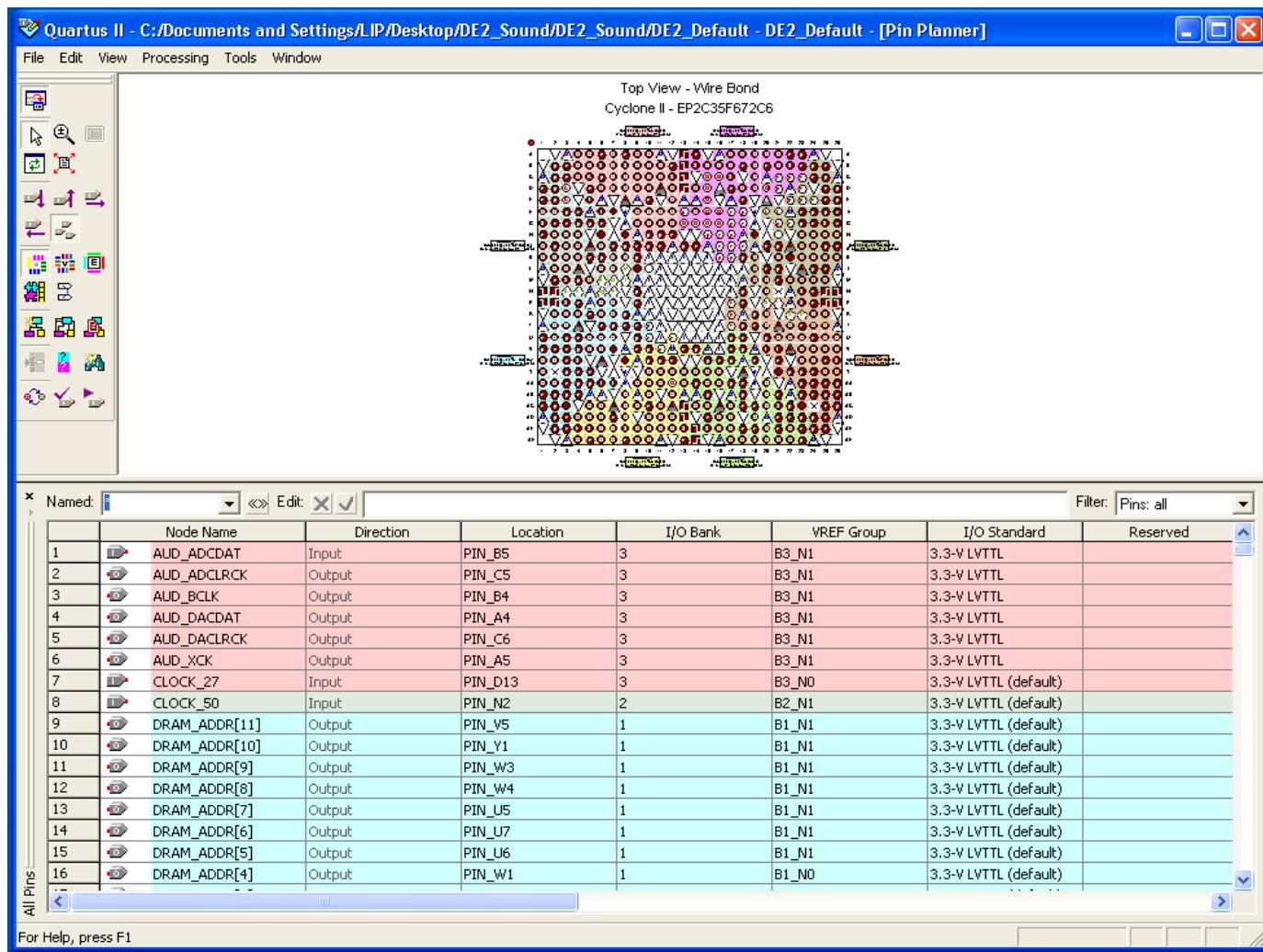
BRAM_W_N	PIN_A010	SRAM Write Enable
BRAM_CE_N	PIN_A010	SRAM Output Enable
BRAM_CS_N	PIN_A010	SRAM Chip Select
BRAM_LB_N	PIN_A009	SRAM Load/Write Data Mask
BRAM_CE_N	PIN_A011	SRAM Chip Enable
OTD_ADDRESS	PIN_F7	I2P132 Address[7]
OTD_ADDRESS	PIN_F2	I2P132 Address[6]
OTD_ADDRESS	PIN_F1	I2P132 Address[5]
OTD_DATA[1]	PIN_E8	I2P132 Data[1]
OTD_DATA[0]	PIN_E9	I2P132 Data[0]
OTD_DATA[3]	PIN_F7	I2P132 Data[3]
OTD_DATA[2]	PIN_J5	I2P132 Data[2]
OTD_DATA[1]	PIN_J6	I2P132 Data[1]
OTD_DATA[0]	PIN_J7	I2P132 Data[0]
OTD_DATA[3]	PIN_H8	I2P132 Data[3]
OTD_DATA[2]	PIN_H9	I2P132 Data[2]
OTD_DATA[1]	PIN_H10	I2P132 Data[1]
OTD_DATA[0]	PIN_H11	I2P132 Data[0]
OTD_DATA[3]	PIN_E1	I2P132 Data[3]
OTD_DATA[2]	PIN_E2	I2P132 Data[2]
OTD_DATA[1]	PIN_E3	I2P132 Data[1]
OTD_DATA[0]	PIN_E4	I2P132 Data[0]
OTD_DATA[3]	PIN_G4	I2P132 Data[3]
OTD_DATA[2]	PIN_G3	I2P132 Data[2]
OTD_DATA[1]	PIN_G2	I2P132 Data[1]
OTD_DATA[0]	PIN_G1	I2P132 Data[0]
OTD_GND_N	PIN_H12	I2P132 Chip Select
OTD_VDD_N	PIN_H13	I2P132 Power
OTD_WR_N	PIN_G1	I2P132 Write
OTD_RST_N	PIN_G5	I2P132 Reset
OTD_CS_N	PIN_G6	I2P132 Chip Select
OTD_IN1	PIN_C3	I2P132 Interrupt 1
OTD_IN2	PIN_C4	I2P132 Interrupt 2
OTD_DACK_N	PIN_B2	I2P132 DMA Acknowledge 1
OTD_DREQ_N	PIN_F8	I2P132 DMA Request 1
OTD_DACK_N	PIN_B3	I2P132 DMA Acknowledge 2
OTD_DREQ_N	PIN_F9	I2P132 DMA Request 2
OTD_FPSPEED	PIN_F3	USB Full Speed, 0 = Enable, 1 = Disable
OTD_LIPSEED	PIN_G8	USB Low Speed, 0 = Enable, 1 = Disable
LCD_DATA[0]	PIN_A1	LCD Data[0]
LCD_DATA[1]	PIN_B2	LCD Data[1]
LCD_DATA[2]	PIN_C3	LCD Data[2]
LCD_DATA[3]	PIN_D4	LCD Data[3]
LCD_DATA[4]	PIN_E5	LCD Data[4]
LCD_DATA[5]	PIN_F6	LCD Data[5]
LCD_DATA[6]	PIN_G7	LCD Data[6]
LCD_DATA[7]	PIN_H8	LCD Data[7]
LCD_VDD_N	PIN_A4	LCD Read/Write Control, 0 = Read, 1 = Write
LCD_BN	PIN_K5	LCD Enable
LCD_CS_N	PIN_L4	LCD Command and Data Command, 1 = Data
LCD_BLON	PIN_K2	LCD Back Light On/Off
SD_SCK	PIN_A004	SD Card Data
SD_SDAT	PIN_A004	SD Card Data
SD_SDAT	PIN_A023	SD Card Data 3
SD_WB	PIN_A023	SD Card Write Blanking
SD_CLK	PIN_A025	SD Card Clock
Signal Name	PIN_Pins No	PIN_Pins No
TCS	PIN_A14	CPLD - FPGAs Ports (A)
TCK	PIN_A14	CPLD - FPGAs Ports (C)
TDI	PIN_D14	CPLD - FPGAs Ports (B)
TDO	PIN_D14	FPGA - CPLD Port B/E/S
Signal Name	PIN_Pins No	PIN_Pins No
INDX_IN	PIN_A024	PCI Transmitter
DATA_IN	PIN_A024	PCI Receiver
Signal Name	PIN_Pins No	PIN_Pins No

Altera DE2 Board Pin Table

GPO_#1 0	PIN_#18	GPO Connection #1 0
GPO_#1 1	PIN_#19	GPO Connection #1 1
GPO_#1 2	PIN_#20	GPO Connection #1 2
GPO_#1 3	PIN_#21	GPO Connection #1 3
GPO_#1 4	PIN_#22	GPO Connection #1 4
GPO_#1 5	PIN_#23	GPO Connection #1 5
GPO_#1 6	PIN_#24	GPO Connection #1 6
GPO_#1 7	PIN_#25	GPO Connection #1 7
GPO_#1 8	PIN_#26	GPO Connection #1 8
GPO_#1 9	PIN_#27	GPO Connection #1 9
GPO_#1 10	PIN_#28	GPO Connection #1 10
GPO_#1 11	PIN_#29	GPO Connection #1 11
GPO_#1 12	PIN_#30	GPO Connection #1 12
GPO_#1 13	PIN_#31	GPO Connection #1 13
GPO_#1 14	PIN_#32	GPO Connection #1 14
GPO_#1 15	PIN_#33	GPO Connection #1 15
GPO_#1 16	PIN_#34	GPO Connection #1 16
GPO_#1 17	PIN_#35	GPO Connection #1 17
GPO_#1 18	PIN_#36	GPO Connection #1 18
GPO_#1 19	PIN_#37	GPO Connection #1 19
GPO_#1 20	PIN_#38	GPO Connection #1 20
GPO_#1 21	PIN_#39	GPO Connection #1 21
GPO_#1 22	PIN_#40	GPO Connection #1 22
GPO_#1 23	PIN_#41	GPO Connection #1 23
GPO_#1 24	PIN_#42	GPO Connection #1 24
GPO_#1 25	PIN_#43	GPO Connection #1 25
GPO_#1 26	PIN_#44	GPO Connection #1 26
GPO_#1 27	PIN_#45	GPO Connection #1 27
GPO_#1 28	PIN_#46	GPO Connection #1 28
GPO_#1 29	PIN_#47	GPO Connection #1 29
GPO_#1 30	PIN_#48	GPO Connection #1 30
GPO_#1 31	PIN_#49	GPO Connection #1 31
GPO_#1 32	PIN_#50	GPO Connection #1 32
GPO_#1 33	PIN_#51	GPO Connection #1 33
GPO_#2 0	PIN_#18	GPO Connection #2 0
GPO_#2 1	PIN_#19	GPO Connection #2 1
GPO_#2 2	PIN_#20	GPO Connection #2 2
GPO_#2 3	PIN_#21	GPO Connection #2 3
GPO_#2 4	PIN_#22	GPO Connection #2 4
GPO_#2 5	PIN_#23	GPO Connection #2 5
GPO_#2 6	PIN_#24	GPO Connection #2 6
GPO_#2 7	PIN_#25	GPO Connection #2 7
GPO_#2 8	PIN_#26	GPO Connection #2 8
GPO_#2 9	PIN_#27	GPO Connection #2 9
GPO_#2 10	PIN_#28	GPO Connection #2 10
GPO_#2 11	PIN_#29	GPO Connection #2 11
GPO_#2 12	PIN_#30	GPO Connection #2 12
GPO_#2 13	PIN_#31	GPO Connection #2 13
GPO_#2 14	PIN_#32	GPO Connection #2 14
GPO_#2 15	PIN_#33	GPO Connection #2 15
GPO_#2 16	PIN_#34	GPO Connection #2 16
GPO_#2 17	PIN_#35	GPO Connection #2 17
GPO_#2 18	PIN_#36	GPO Connection #2 18
GPO_#2 19	PIN_#37	GPO Connection #2 19
GPO_#2 20	PIN_#38	GPO Connection #2 20
GPO_#2 21	PIN_#39	GPO Connection #2 21
GPO_#2 22	PIN_#40	GPO Connection #2 22
GPO_#2 23	PIN_#41	GPO Connection #2 23
GPO_#2 24	PIN_#42	GPO Connection #2 24
GPO_#2 25	PIN_#43	GPO Connection #2 25
GPO_#2 26	PIN_#44	GPO Connection #2 26
GPO_#2 27	PIN_#45	GPO Connection #2 27
GPO_#2 28	PIN_#46	GPO Connection #2 28
GPO_#2 29	PIN_#47	GPO Connection #2 29
GPO_#2 30	PIN_#48	GPO Connection #2 30
GPO_#2 31	PIN_#49	GPO Connection #2 31
GPO_#2 32	PIN_#50	GPO Connection #2 32
GPO_#2 33	PIN_#51	GPO Connection #2 33
GPO_#3 0	PIN_#18	GPO Connection #3 0
GPO_#3 1	PIN_#19	GPO Connection #3 1
GPO_#3 2	PIN_#20	GPO Connection #3 2
GPO_#3 3	PIN_#21	GPO Connection #3 3
GPO_#3 4	PIN_#22	GPO Connection #3 4
GPO_#3 5	PIN_#23	GPO Connection #3 5
GPO_#3 6	PIN_#24	GPO Connection #3 6
GPO_#3 7	PIN_#25	GPO Connection #3 7
GPO_#3 8	PIN_#26	GPO Connection #3 8
GPO_#3 9	PIN_#27	GPO Connection #3 9
GPO_#3 10	PIN_#28	GPO Connection #3 10
GPO_#3 11	PIN_#29	GPO Connection #3 11
GPO_#3 12	PIN_#30	GPO Connection #3 12
GPO_#3 13	PIN_#31	GPO Connection #3 13
GPO_#3 14	PIN_#32	GPO Connection #3 14
GPO_#3 15	PIN_#33	GPO Connection #3 15
GPO_#3 16	PIN_#34	GPO Connection #3 16
GPO_#3 17	PIN_#35	GPO Connection #3 17
GPO_#3 18	PIN_#36	GPO Connection #3 18
GPO_#3 19	PIN_#37	GPO Connection #3 19
GPO_#3 20	PIN_#38	GPO Connection #3 20
GPO_#3 21	PIN_#39	GPO Connection #3 21
GPO_#3 22	PIN_#40	GPO Connection #3 22
GPO_#3 23	PIN_#41	GPO Connection #3 23
GPO_#3 24	PIN_#42	GPO Connection #3 24
GPO_#3 25	PIN_#43	GPO Connection #3 25
GPO_#3 26	PIN_#44	GPO Connection #3 26
GPO_#3 27	PIN_#45	GPO Connection #3 27
GPO_#3 28	PIN_#46	GPO Connection #3 28
GPO_#3 29	PIN_#47	GPO Connection #3 29
GPO_#3 30	PIN_#48	GPO Connection #3 30
GPO_#3 31	PIN_#49	GPO Connection #3 31
GPO_#3 32	PIN_#50	GPO Connection #3 32
GPO_#3 33	PIN_#51	GPO Connection #3 33

161

17]



30 seconds to light a LED

File→Open Project→DE2_top

```
// V1.2 :| Johnny Chen :| 05/11/16 :| Fixed ISP1362 INT/
module DE2_TOP
    // Clock Input
    input CLOCK_27; // 27 MHz
    input CLOCK_50; // 50 MHz
    input EXT_CLOCK; // External Clock
    // Push Button
    input [3:0] KEY; // Pushbutton[3:0]
    // DPDT Switch
    input [17:0] SW; // Toggle Switch[17:0]
    // 7-SEG Dispaly
    output [6:0] HEX0; // Seven Segment Digit 0
    output [6:0] HEX1; // Seven Segment Digit 1
    output [6:0] HEX2; // Seven Segment Digit 2
    output [6:0] HEX3; // Seven Segment Digit 3
    output [6:0] HEX4; // Seven Segment Digit 4
    output [6:0] HEX5; // Seven Segment Digit 5
    output [6:0] HEX6; // Seven Segment Digit 6
    output [6:0] HEX7; // Seven Segment Digit 7
    // LED
    output [8:0] LEDG; // LED Green[8:0]
    output [17:0] LEDR; // LED Red[17:0]
    // UART
    output UART_TXD; // UART Transmitter
    input  UART_RXD; // UART Receiver
    // IRDA
    output IRDA_TXD; // IRDA Transmitter
    input  IRDA_RXD; // IRDA Receiver
    // SDRAM Interface
    inout [15:0] DRAM_DQ; // SDRAM Data bus 16 Bits
    output [11:0] DRAM_ADDR; // SDRAM Address bus 12 Bits

    // Turn on all display
    assign HEX0 = 7'h00;
    assign HEX1 = 7'h00;
    assign HEX2 = 7'h00;
    assign HEX3 = 7'h00;
    assign HEX4 = 7'h00;
    assign HEX5 = 7'h00;
    assign HEX6 = 7'h00;
    assign HEX7 = 7'h00;
    assign LEDG = 9'h1FFF;
    assign LEDR = 18'h3FFFF;
    assign LCD_ON = 1'b1;
    assign LCD_BLON = 1'b1;

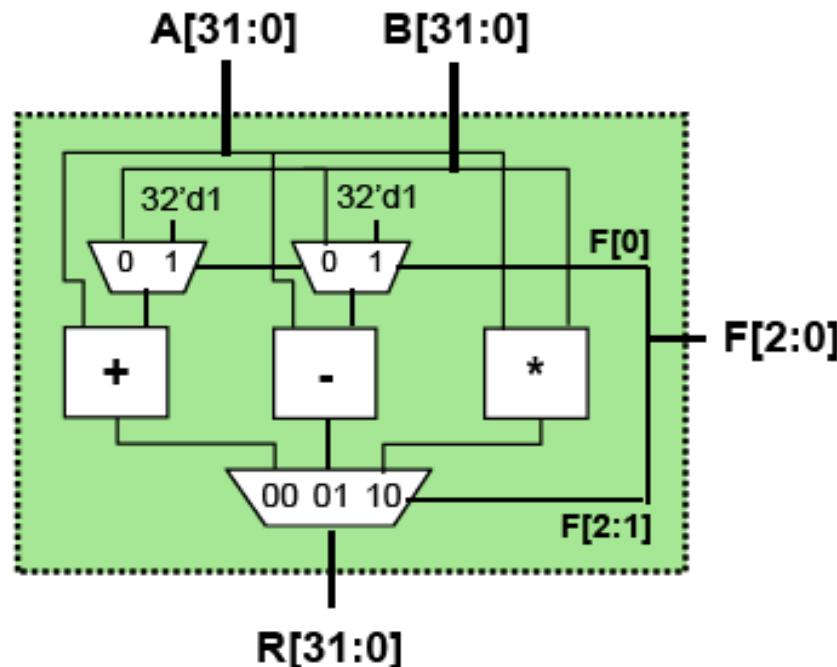
    // All inout port turn to tri-state
    assign DRAM_DQ = 16'hzzzz;
    assign FL_DQ = 8'hzz;
    assign SRAM_DQ = 16'hzzzz;
    assign OTG_DATA = 16'hzzzz;
    assign LCD_DATA = 8'hzz;
    assign SD_DAT = 1'bzz;
    assign I2C_SDAT = 1'bzz;
    assign ENET_DATA = 16'hzzzz;
    assign AUD_ADLRCK = 1'bzz;
    assign AUD_DACLRCK = 1'bzz;
    assign AUD_BCLK = 1'bzz;
    assign GPIO_0 = 36'hzzzzzzzz;
    assign GPIO_1 = 36'hzzzzzzzz;
```

Build your own computer...

Connecting modules to build complex machines

- Modularity is essential to the success of large designs
- A Verilog module may contain submodules that are “wired together”
- High-level primitives enable direct synthesis of behavioral descriptions (functions such as additions, subtractions, shifts (`<<` and `>>`), etc.)

Example: A 32-bit ALU



Function Table

F2	F1	F0	Function
0	0	0	A + B
0	0	1	A + 1
0	1	0	A - B
0	1	1	A - 1
1	0	X	A * B

Modules

2-to-1 MUX

```
module mux32two(i0,i1,sel,out);
  input [31:0] i0,i1;
  input sel;
  output [31:0] out;

  assign out = sel ? i1 : i0;

endmodule
```

3-to-1 MUX

```
module mux32three(i0,i1,i2,sel,out);
  input [31:0] i0,i1,i2;
  input [1:0] sel;
  output [31:0] out;
  reg [31:0] out;

  always @ (i0 or i1 or i2 or sel)
    begin
      case (sel)
        2'b00: out = i0;
        2'b01: out = i1;
        2'b10: out = i2;
        default: out = 32'bx;
      endcase
    end
  endmodule
```

32-bit Adder

```
module add32(i0,i1,sum);
  input [31:0] i0,i1;
  output [31:0] sum;

  assign sum = i0 + i1;

endmodule
```

32-bit Subtractor

```
module sub32(i0,i1,diff);
  input [31:0] i0,i1;
  output [31:0] diff;

  assign diff = i0 - i1;

endmodule
```

16-bit Multiplier

```
module mul16(i0,i1,prod);
  input [15:0] i0,i1;
  output [31:0] prod;

  // this is a magnitude multiplier
  // signed arithmetic later
  assign prod = i0 * i1;

endmodule
```

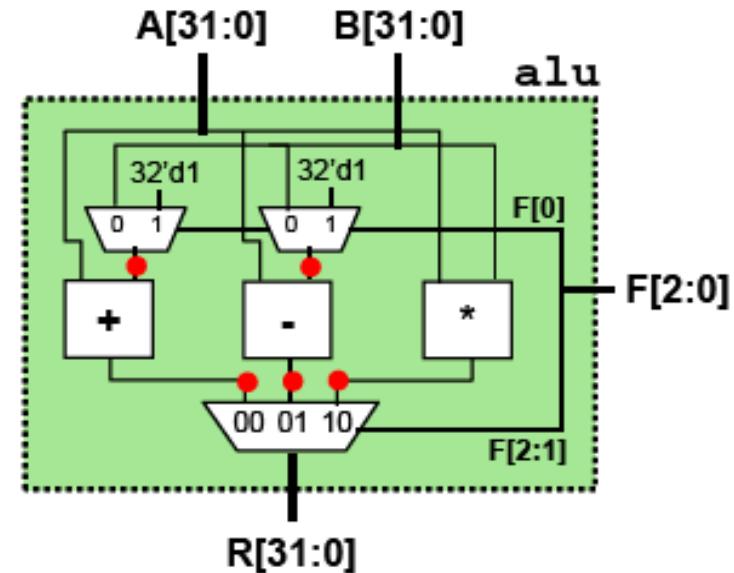
Top-Level: connect the modules

Given submodules:

```
module mux32two(10,11,sel,out);  
module mux32three(10,11,12,sel,out);  
module add32(10,11,sum);  
module sub32(10,11,diff);  
module mul16(10,11,prod);
```

Declaration of the ALU Module:

```
module alu(a, b, f, r);  
    input [31:0] a, b;  
    input [2:0] f;  
    output [31:0] r;  
  
    wire [31:0] addmux_out, submux_out;  
    wire [31:0] add_out, sub_out, mul_out;  
  
    mux32two adder_mux(b, 32'd1, f[0], addmux_out);  
    mux32two sub_mux(b, 32'd1, f[0], submux_out);  
    add32 our_adder(a, addmux_out, add_out);  
    sub32 our_subtracter(a, submux_out, sub_out);  
    mul16 our_multiplier(a[15:0], b[15:0], mul_out);  
    mux32three output_mux(add_out, sub_out, mul_out, f[2:1], r);  
  
endmodule
```



intermediate output nodes ●

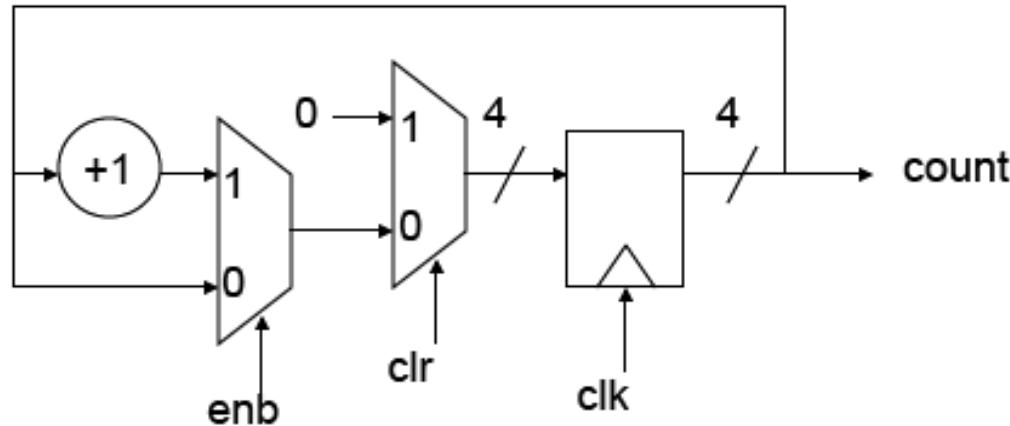
module
names

(unique)
instance
names

corresponding
wires/regs in
module alu

Example: A simple counter

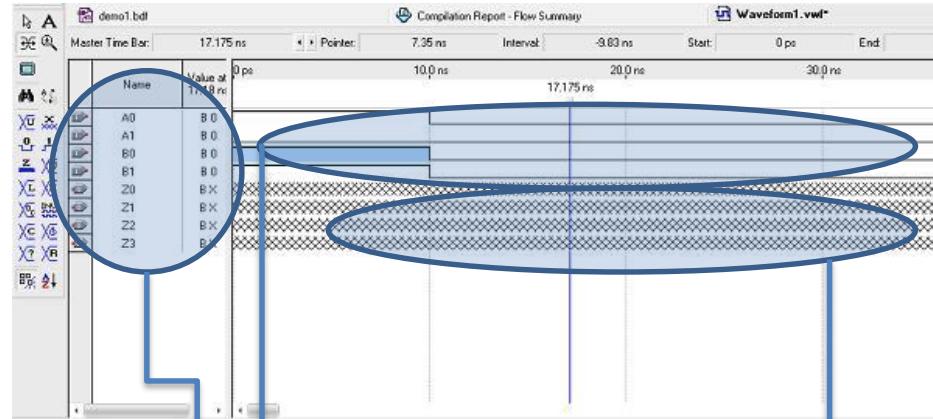
Do you know a simple way to count 10 ns pulses? Lots of them?



```
// 4-bit counter with enable and synchronous clear
module counter(input clk,enb,clr,
                output reg [3:0] count);
    always @(posedge clk) begin
        count <= clr ? 4'b0 : (enb ? count+1 : count);
    end
endmodule
```

Tools - Simulation

Quartus II



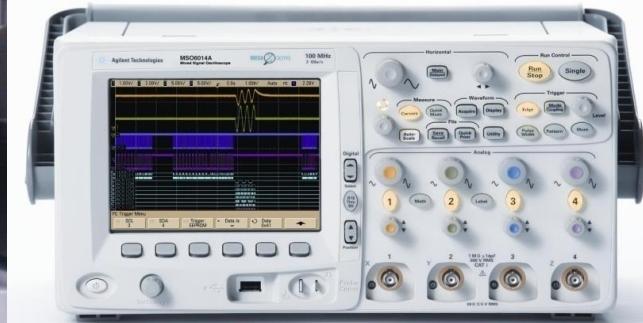
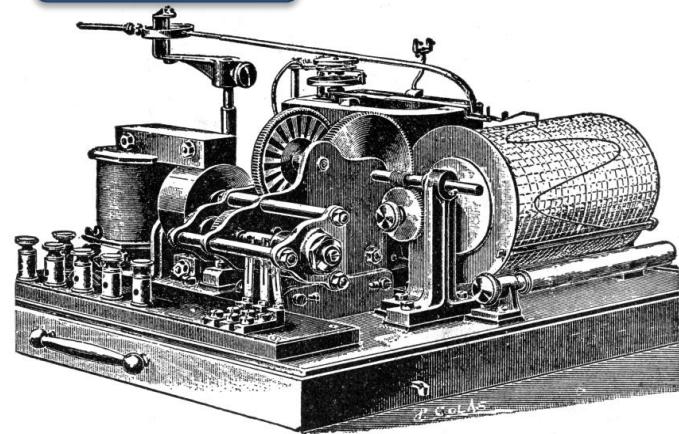
Define the signals :
Inputs – stimulus
Outputs - results

Define the stimulus to the system

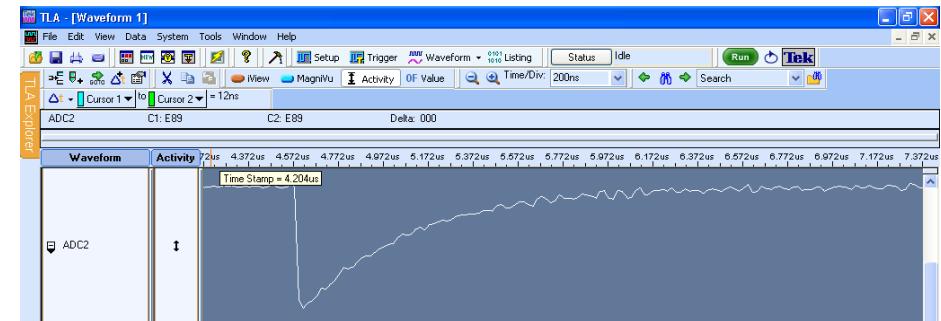
Run the simulation and you get the result

Tools – Measurement instruments

Waveforms



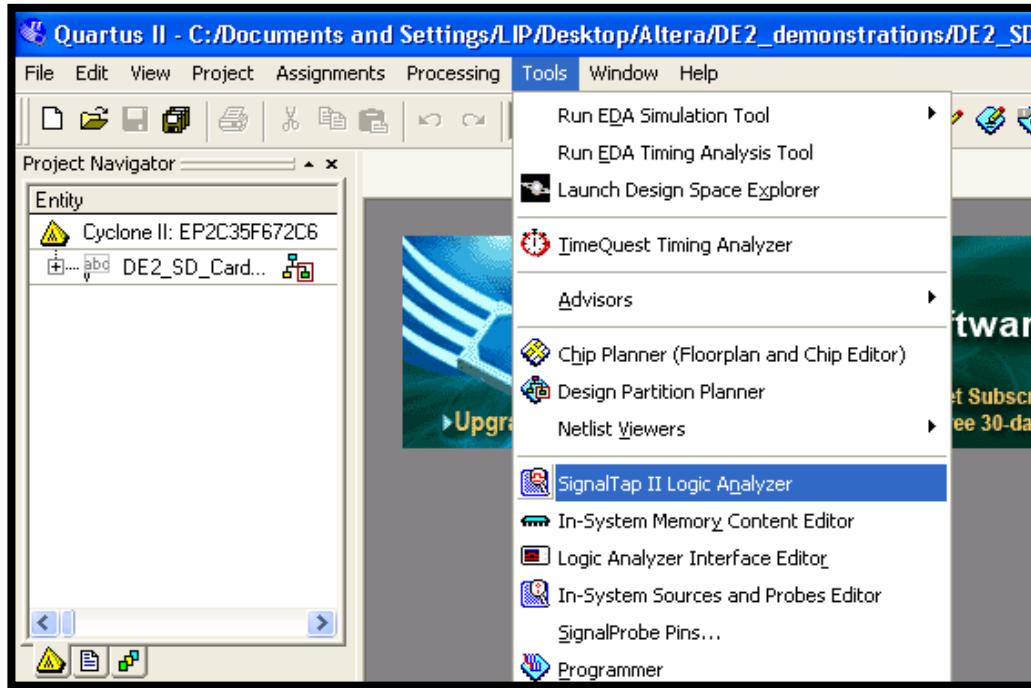
Logic Levels



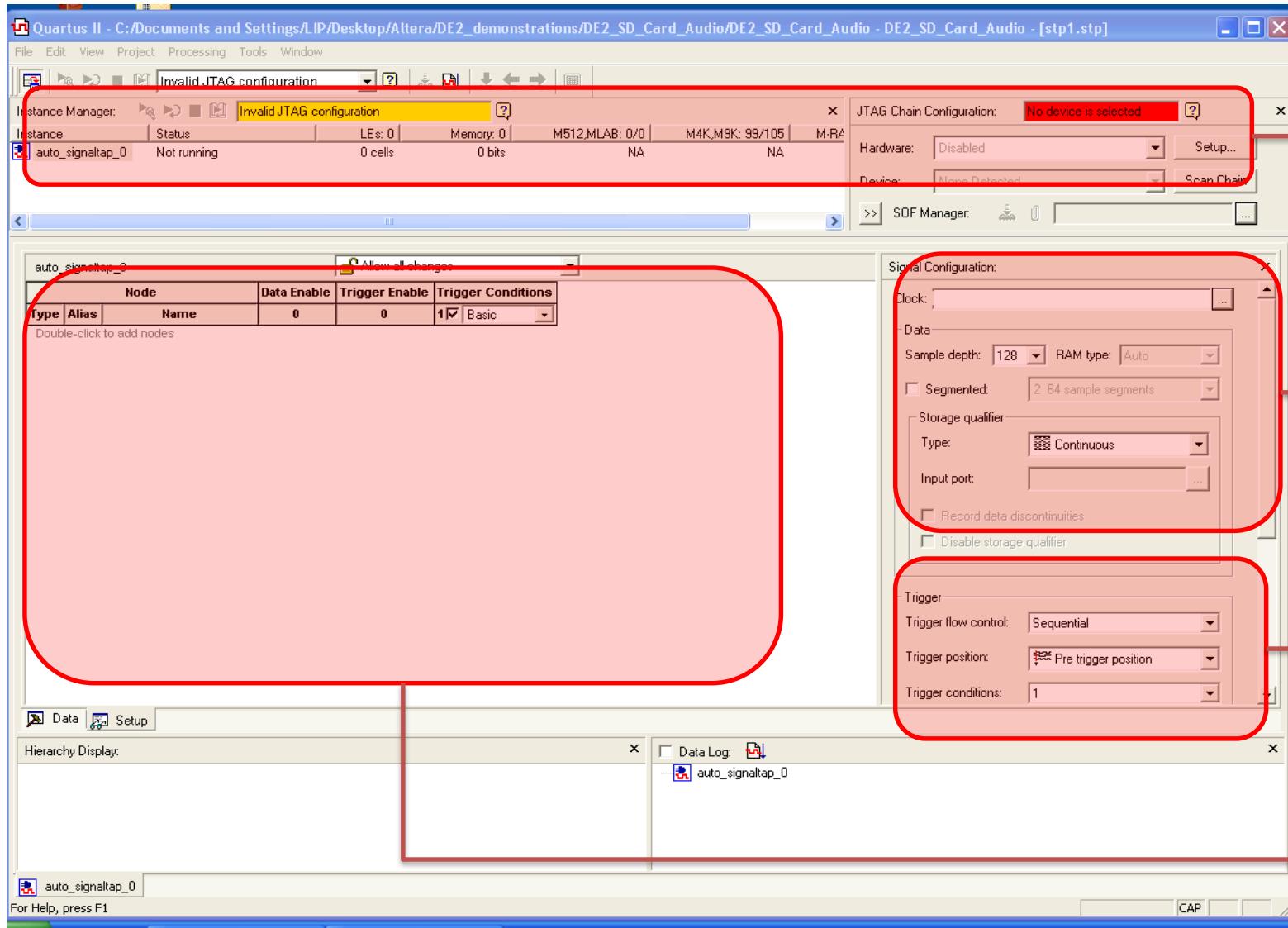
Tools – Internal Logic Analyser

Signal-TAP embedded Logic Analyser

Quartus II Handbook Version 9.0 Volume 3: Verification 14. Design Debugging Using the SignalTap II Embedded Logic Analyzer



Tools – Internal Logic Analyser



CONTROL

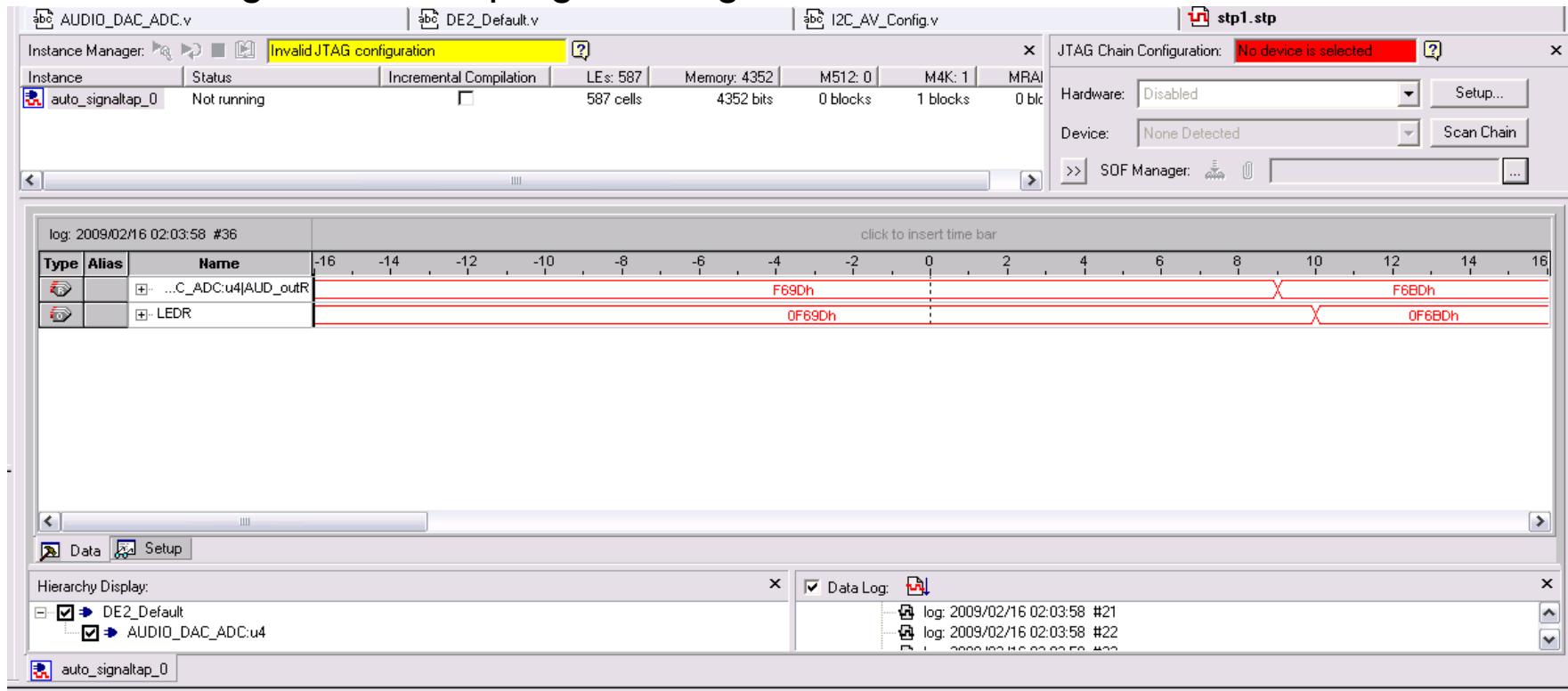
Clock definition

Trigger definition

The signal you want to “see”

Tools – Internal Logic Analyser

The logic analyser will collect data from the registers and output it through the JTAG programming interface

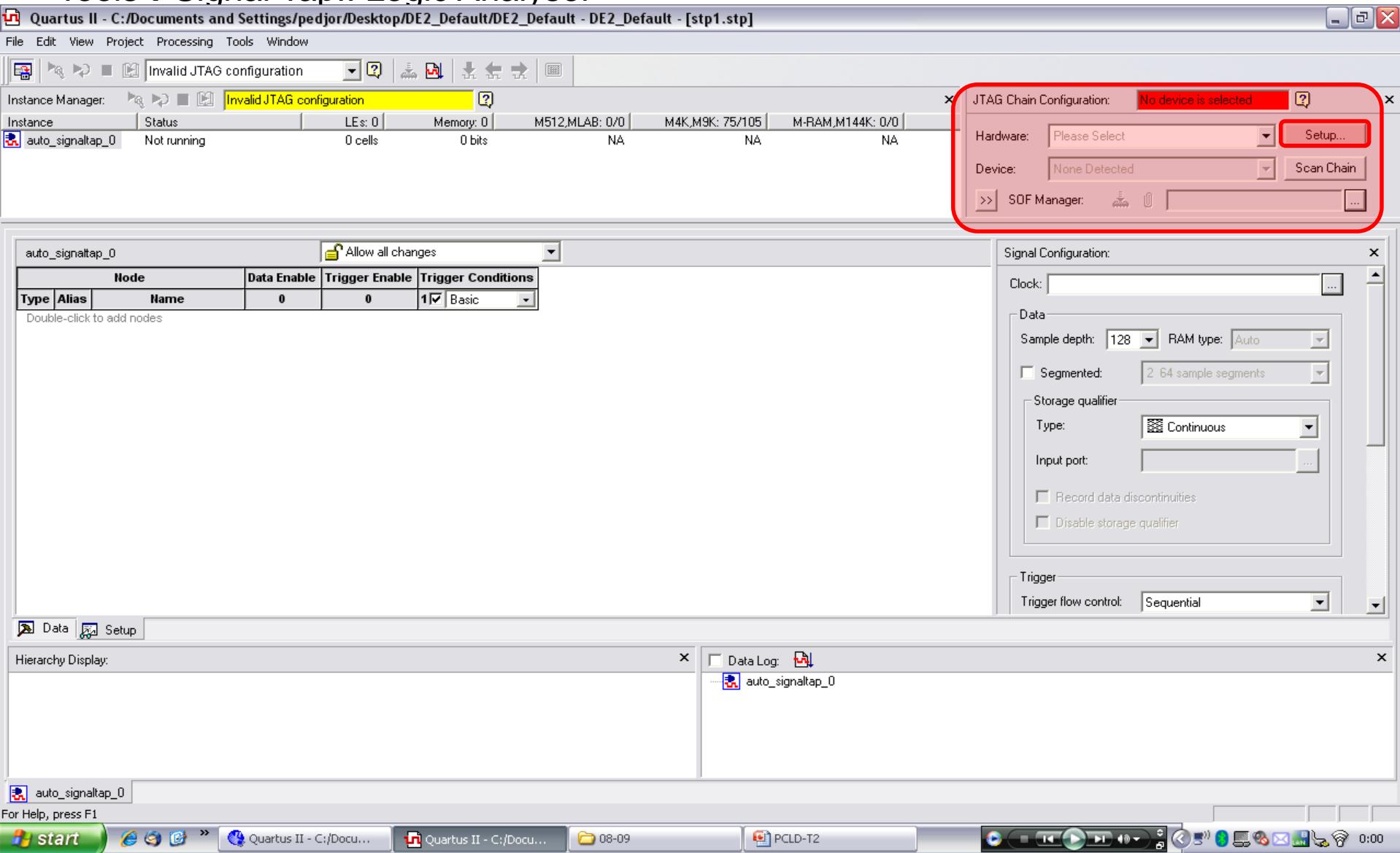


Tools – Signal probe

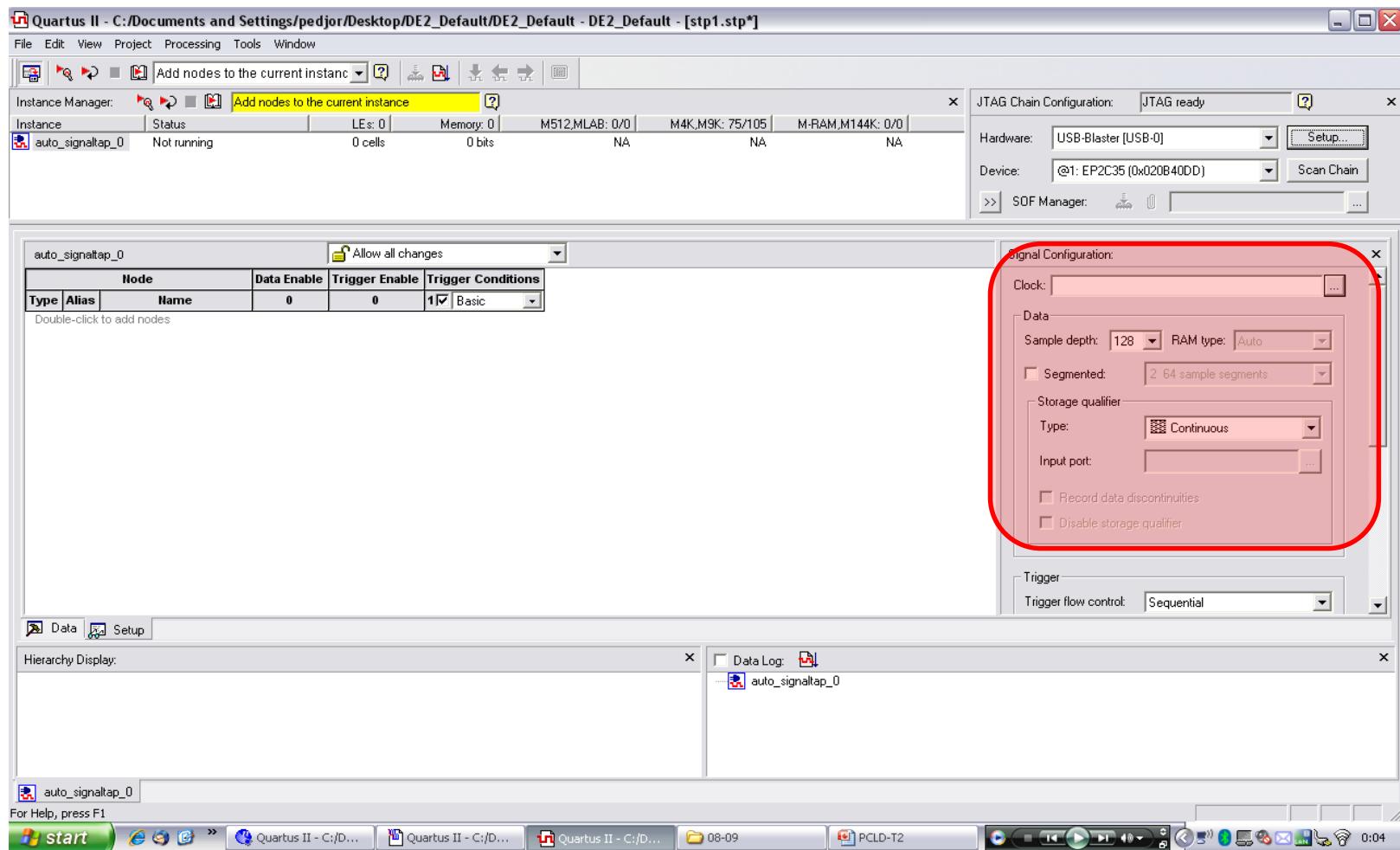
Internal signals can be extracted to output pins and connected to an external logic analyser. Signals can be exchanged easily...

SignalTap step by step

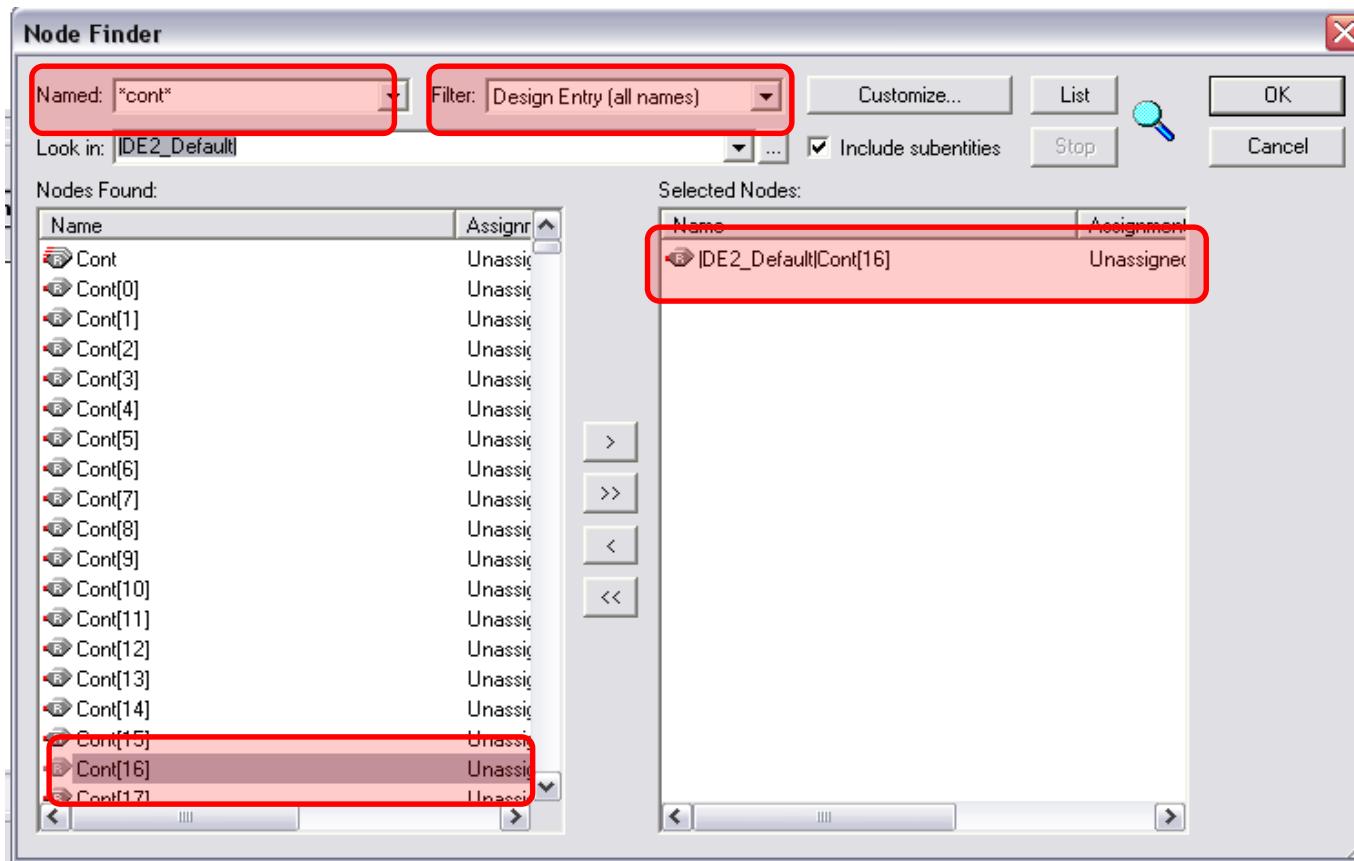
- Open DE2 default and compile it!
- Program DE2
- Tools → Signal Tap II Logic Analyser



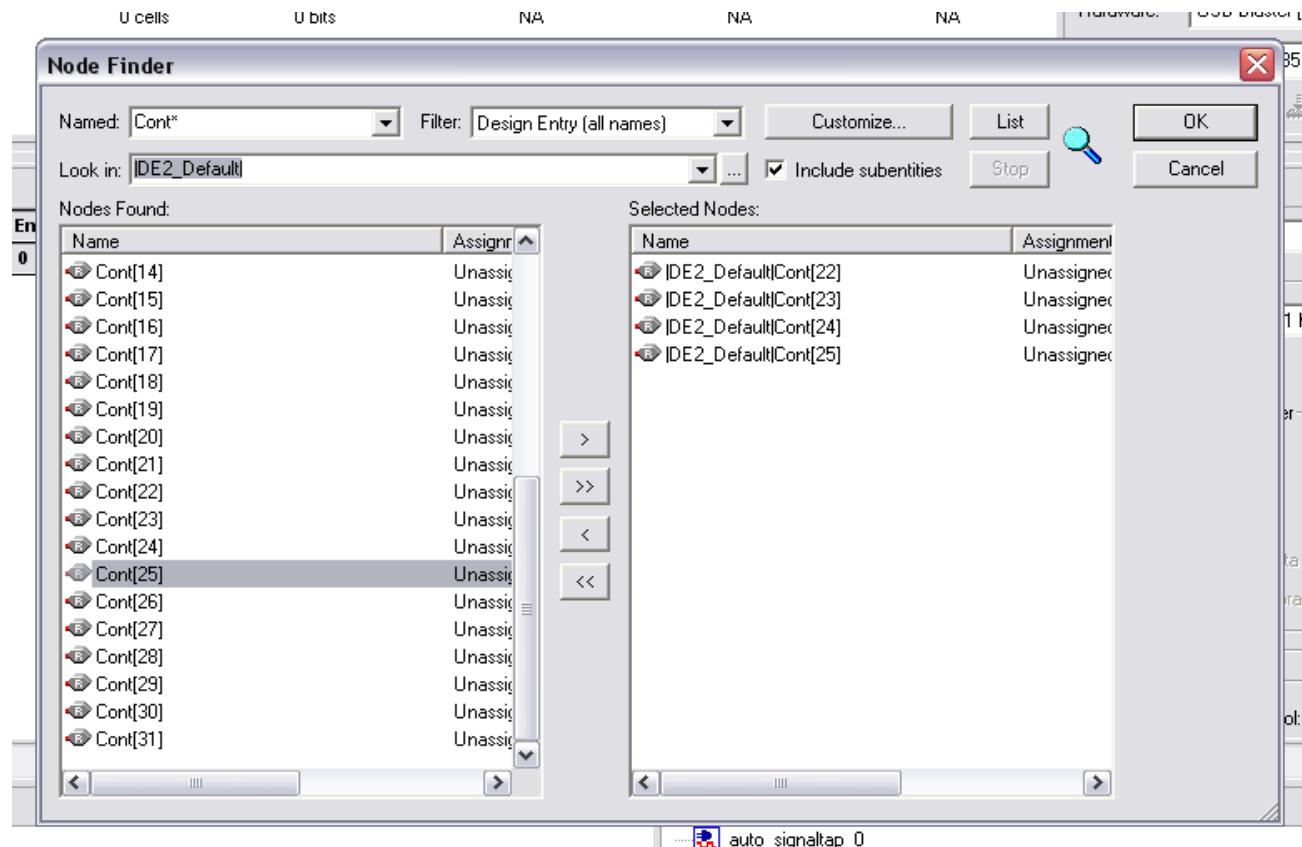
- Setup the hardware (choose the USB blaster)



- Define the clock

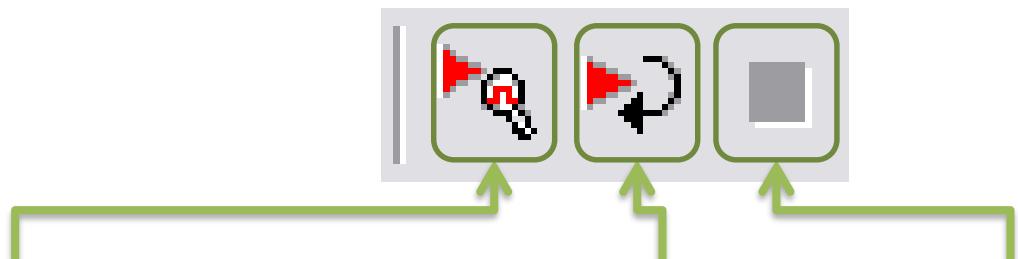
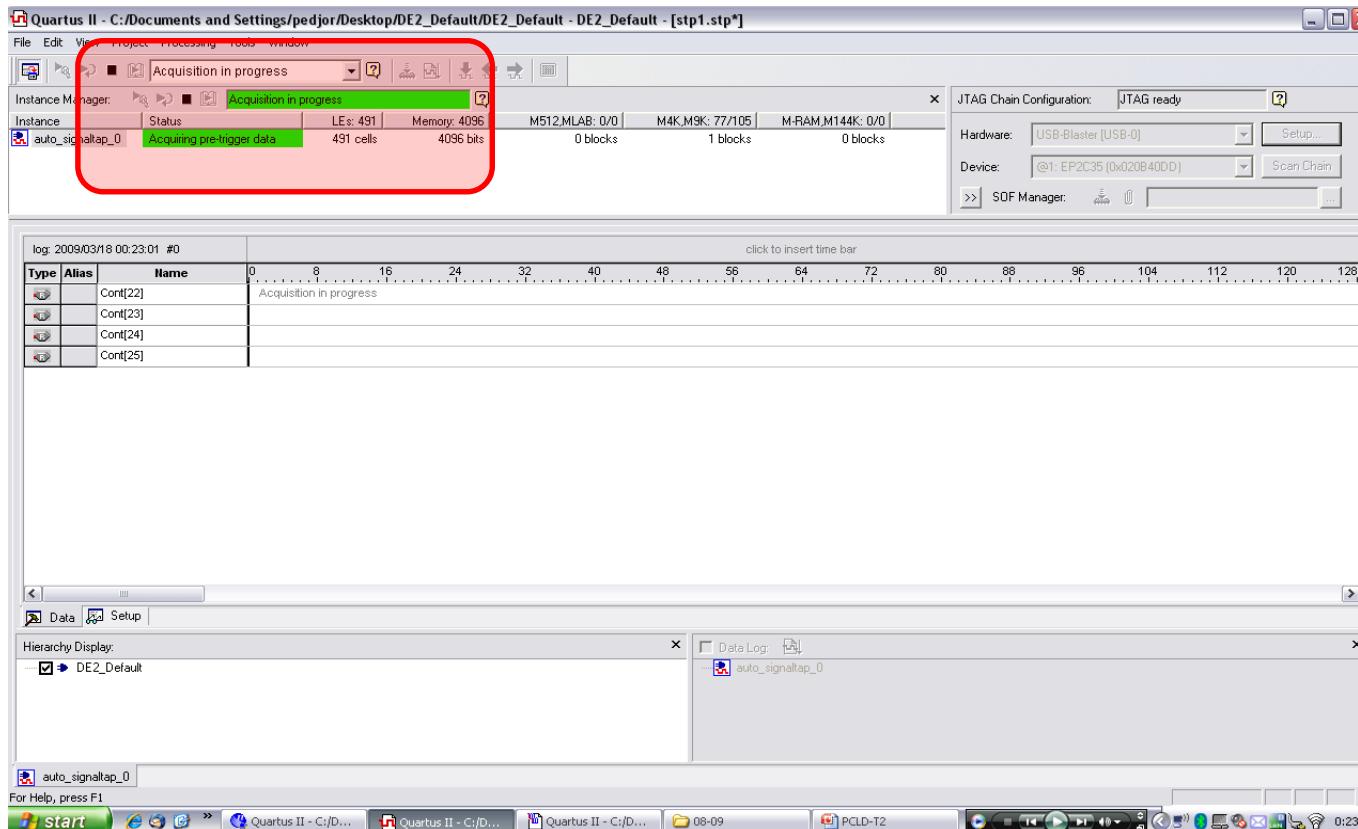


Choose the signals to observe. Choose Cont 25,24,23,22



- Compile the project! You may need to save some files and answer some questions
- Program the board

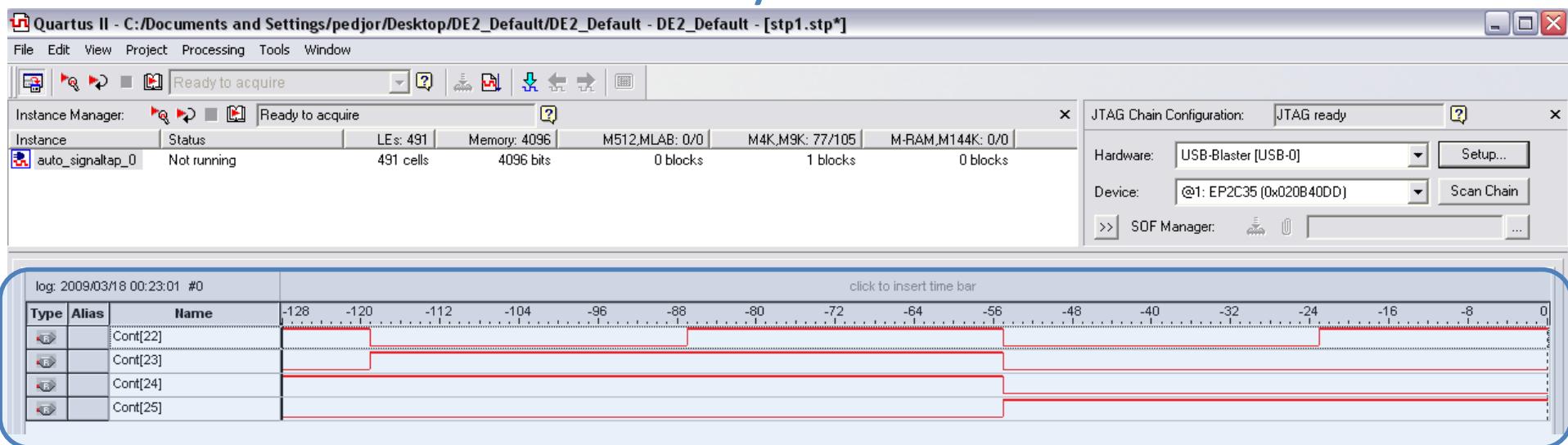
Run Analysis → Green (acquisition in progress, acquiring data)



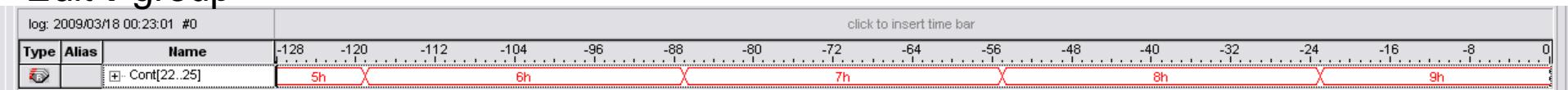
Run Analysis (1 time)

Run Analysis (continuous) Stop

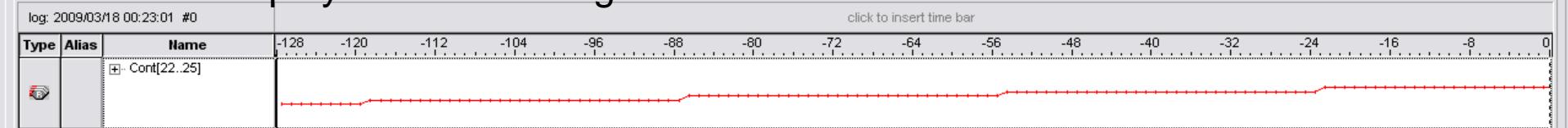
Finally: data!!



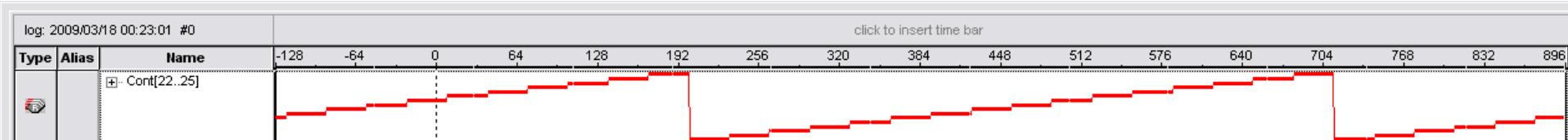
- Select the four signals;
- Edit → group



- Select the group
- Edit → Bus Display Format → Unsigned Line Chart



- Unzoom



Tools – Mega Wizard

Quartus II

File Edit View Project Assignments Processing Tools Window Help

Project Navigator

Entity Logic Cells Dedicated

Hierarchy Files Design Units

Tasks Flow: Compilation

Task Compile Design

+ Analysis & Synthesis

+ Filter (Place & Route)

+ Assembler (Generate programming file)

+ Classic Timing Analysis

+ EDA Netlist Writer

Program Device (Open Programmer)

Type Message

System Processing Extra Info Info Warning Critical Warning Error Suppressed Flag

Message: Location: Locate

Run EDA Simulation Tool

Run EDA Timing Analysis Tool

Launch EDA Simulation Library Compiler

Launch Design Space Explorer

TimeQuest Timing Analyzer

Advisors

Chip Planner (Floorplan and Chip Editor)

Design Partition Planner

Netlist Viewers

SignalTap II Logic Analyzer

In-System Memory Content Editor

Logic Analyzer Interface Editor

In-System Sources and Probes Editor

SignalProbe Pins...

Programmer

MegaWizard Plug-In Manager...

SOPC Builder

Tcl Scripts...

Customize...

Options...

License Setup...

MegaWizard Plug-In Manager [page 1]

The MegaWizard Plug-In Manager helps you create or modify design files that contain custom variations of megafunctions.

Which action do you want to perform?

Create a new custom megafunction variation

Edit an existing custom megafunction variation

Copy an existing custom megafunction variation

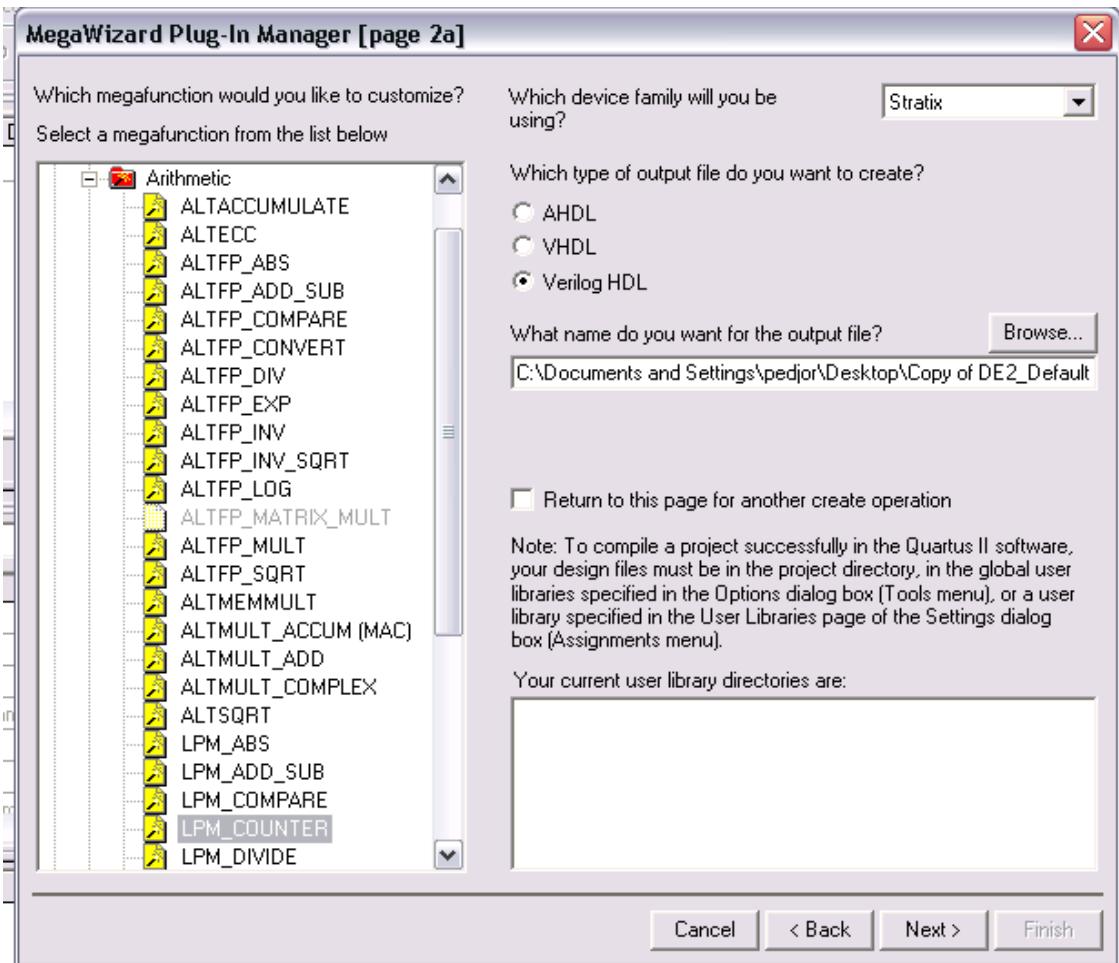
Copyright (C) 1991-2009 Altera Corporation

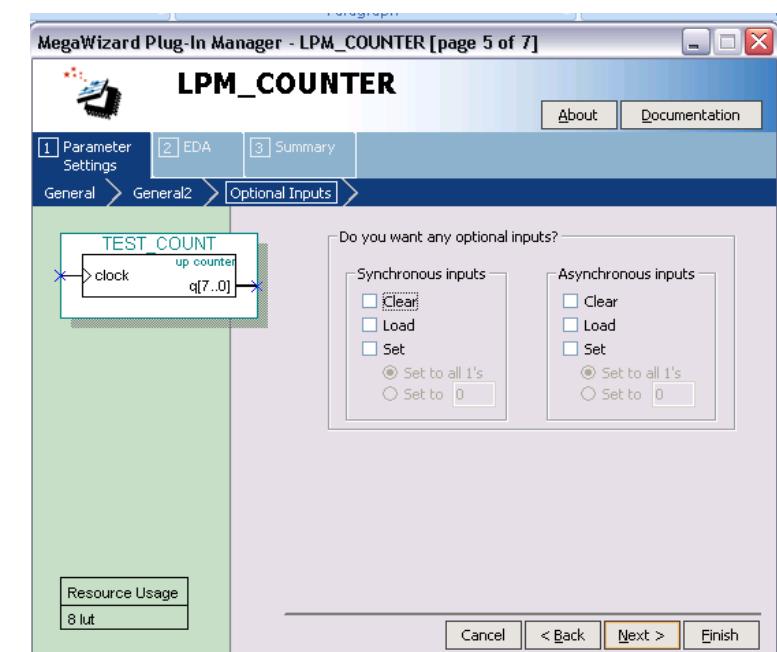
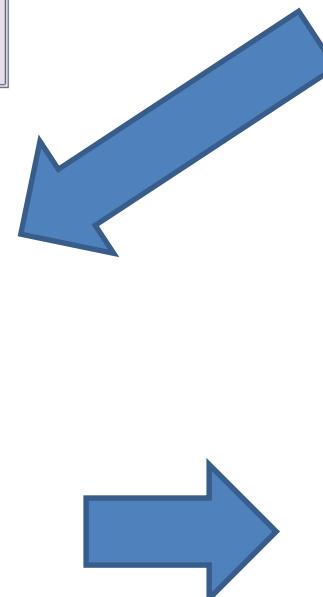
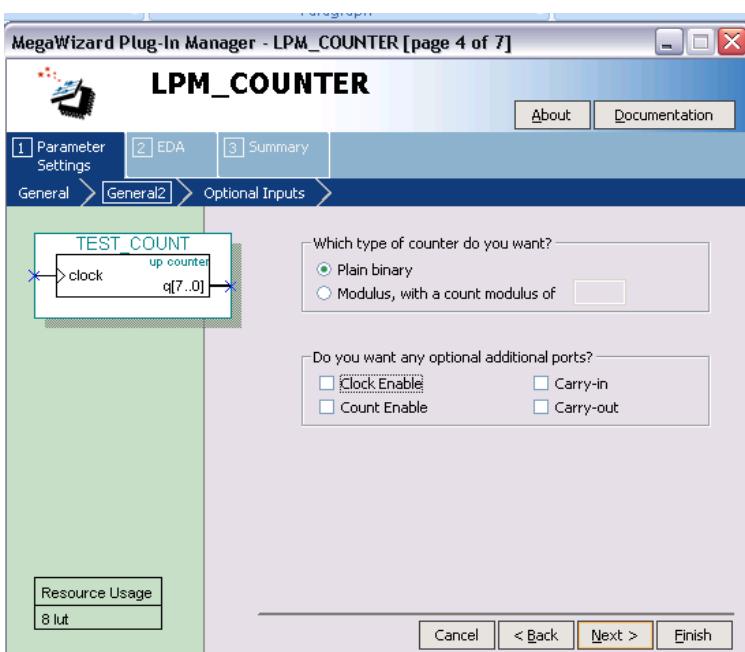
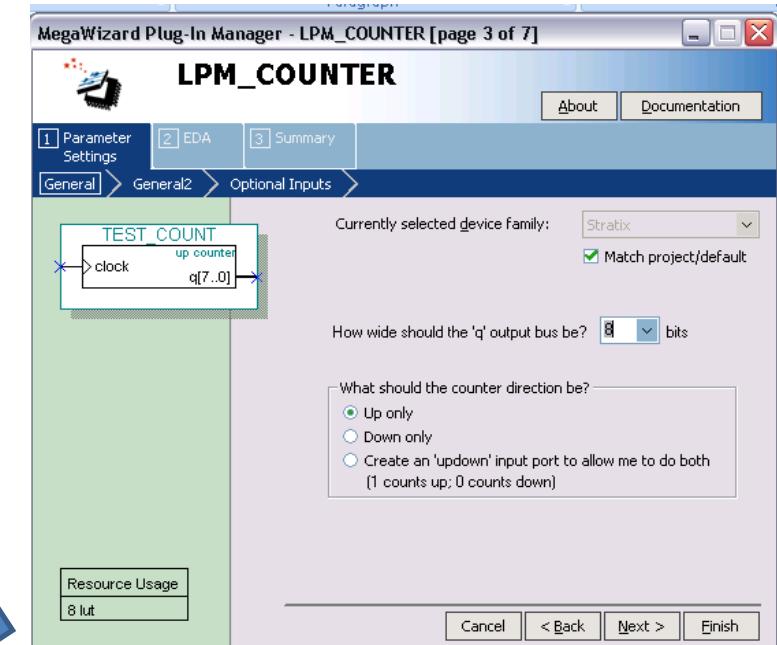
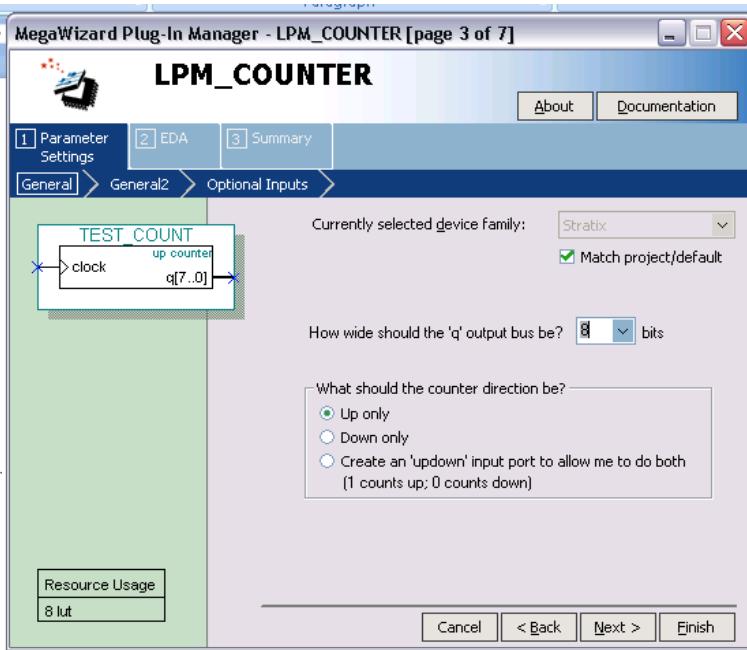
Cancel < Back Next > Finish

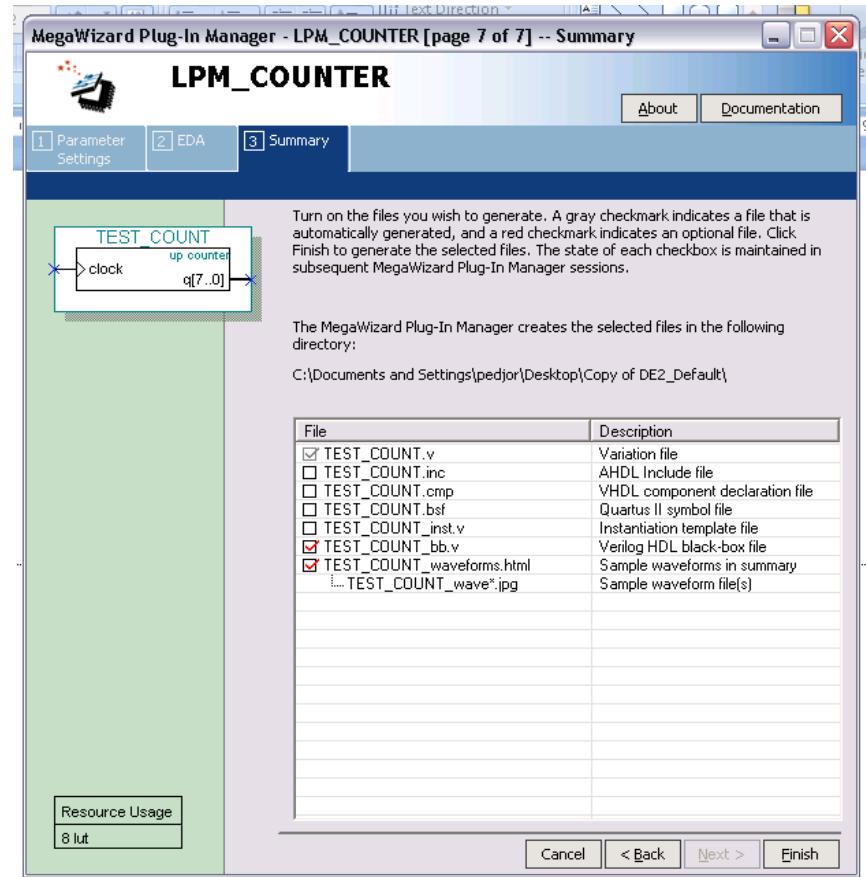
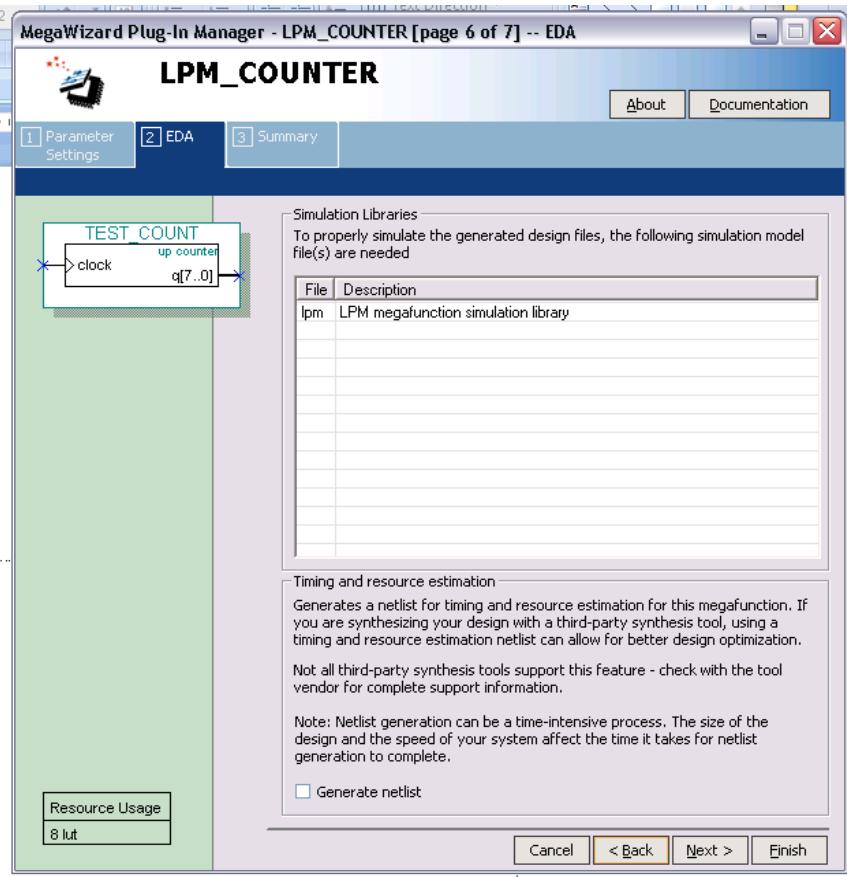
View New Quartus II Information

Documentation

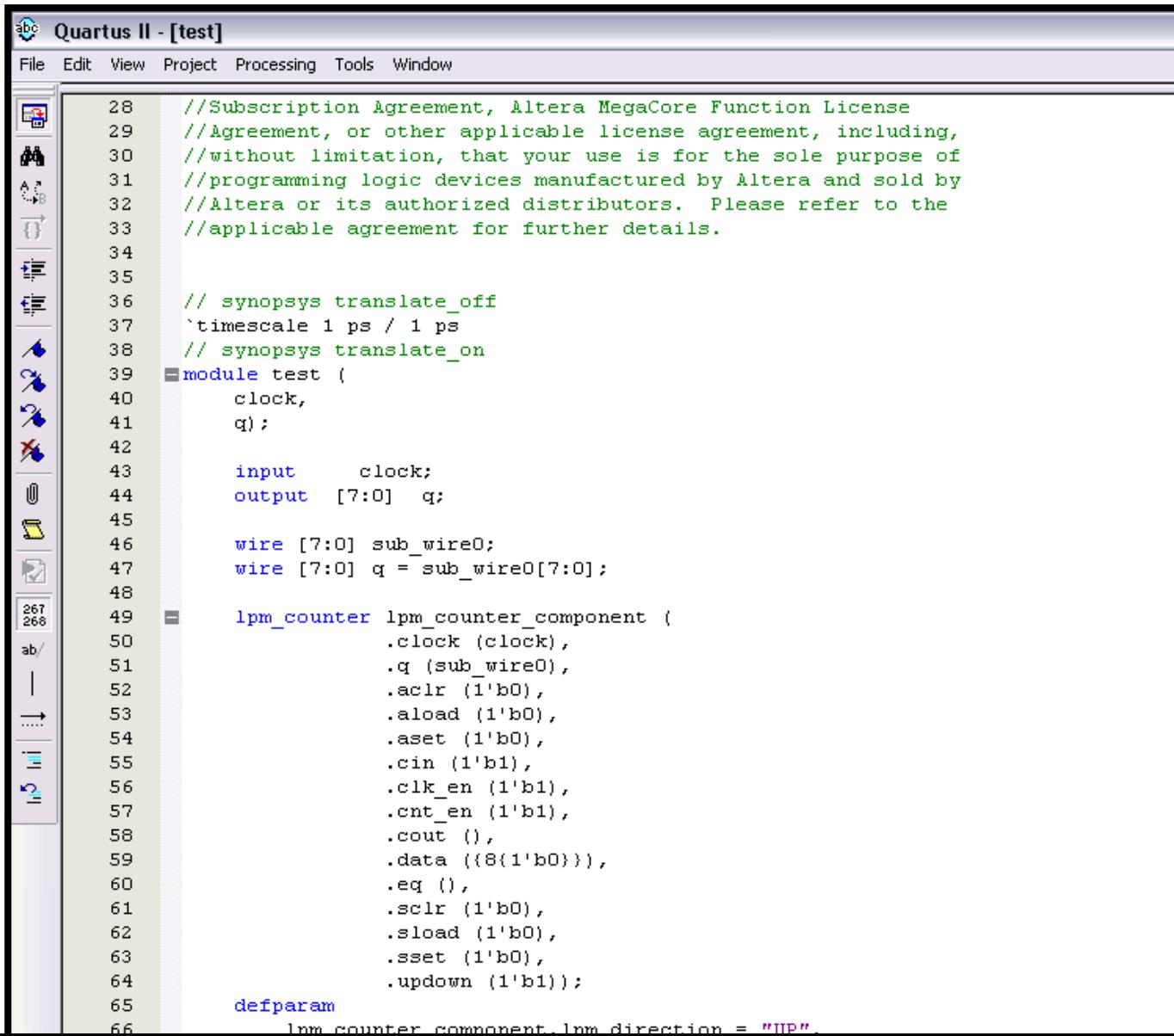
Starts the MegaWizard Plug-In Manager







Mega Wizard – What you get



The screenshot shows the Quartus II software interface with the title bar "Quartus II - [test]". The menu bar includes File, Edit, View, Project, Processing, Tools, and Window. On the left is a toolbar with various icons for file operations like Open, Save, and Find. The main window displays a Verilog code editor with the following content:

```
28 //Subscription Agreement, Altera MegaCore Function License
29 //Agreement, or other applicable license agreement, including,
30 //without limitation, that your use is for the sole purpose of
31 //programming logic devices manufactured by Altera and sold by
32 //Altera or its authorized distributors. Please refer to the
33 //applicable agreement for further details.
34
35
36 // synopsys translate_off
37 `timescale 1 ps / 1 ps
38 // synopsys translate_on
39 module test (
40     clock,
41     q);
42
43     input      clock;
44     output [7:0] q;
45
46     wire [7:0] sub_wire0;
47     wire [7:0] q = sub_wire0[7:0];
48
49     lpm_counter lpm_counter_component (
50         .clock (clock),
51         .q (sub_wire0),
52         .aclr (1'b0),
53         .aload (1'b0),
54         .aset (1'b0),
55         .cin (1'b1),
56         .clk_en (1'b1),
57         .cnt_en (1'b1),
58         .cout (),
59         .data ((8(1'b0))),
60         .eq (),
61         .sclr (1'b0),
62         .sload (1'b0),
63         .sset (1'b0),
64         .updown (1'b1));
65
66     defparam
67         lpm_counter_component.lpm_direction = "UP";
68
69     endmodule
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
687
688
689
689
690
691
692
693
694
695
696
697
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
778
779
779
780
781
782
783
784
785
786
787
787
788
789
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
878
879
879
880
881
882
883
884
885
886
887
887
888
889
889
889
890
891
892
893
894
895
896
897
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
917
918
919
919
920
921
922
923
924
925
926
927
928
928
929
929
930
931
932
933
934
935
936
937
938
938
939
939
940
941
942
943
944
945
946
947
947
948
949
949
950
951
952
953
954
955
956
957
958
958
959
959
960
961
962
963
964
965
966
967
967
968
968
969
969
970
971
972
973
974
975
976
977
977
978
978
979
979
980
981
982
983
984
985
986
987
987
988
988
989
989
990
991
992
993
994
995
996
997
997
998
999
999
1000
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1027
1028
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1037
1038
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1047
1048
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1057
1058
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1091
1092
1093
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1101
1102
1103
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1111
1112
1113
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1121
1122
1123
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1131
1132
1133
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1141
1142
1143
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1151
1152
1153
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1161
1162
1163
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1171
1172
1173
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1181
1182
1183
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1191
1192
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1201
1202
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1211
1212
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1221
1222
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1231
1232
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1241
1242
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1251
1252
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1261
1262
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1271
1272
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1281
1282
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1291
1292
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1301
1302
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1311
1312
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1321
1322
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1331
1332
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1341
1342
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1351
1352
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1361
1362
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1371
1372
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1381
1382
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1391
1392
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1401
1402
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1411
1412
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1421
1422
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1431
1432
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1441
1442
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1451
1452
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1461
1462
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1471
1472
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1481
1482
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1491
1492
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1501
1502
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1511
1512
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1521
1522
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1531
1532
1533
1534
1534
1535
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
1541
1542
1543
1544
1544
1545
1545
1546
1546
1547
1547
1548
1548
1549
1549
1550
1551
1552
1553
1554
1554
1555
1555
1556
1556
1557
1557
1558
1558
1559
1559
1560
1561
1562
1563
1564
1564
1565
1565
1566
1566
1567
1567
1568
1568
1569
1569
1570
1571
1572
1573
1574
1574
1575
1575
1576
1576
1577
1577
1578
1578
1579
1579
1580
1581
1582
1583
1584
1584
1585
1585
1586
1586
1587
1587
1588
1588
1589
1589
1590
1591
1592
1593
1594
1594
1595
1595
1596
1596
1597
1597
1598
1598
1599
1599
1600
1601
1602
1603
1604
1604
1605
1605
1606
1606
1607
1607
1608
1608
1609
1609
1610
1611
1612
1613
1614
1614
1615
1615
1616
1616
1617
1617
1618
1618
1619
1619
1620
1621
1622
1623
1624
1624
1625
1625
1626
1626
1627
1627
1628
1628
1629
1629
1630
1631
1632
1633
1634
1634
1635
1635
1636
1636
1637
1637
1638
1638
1639
1639
1640
1641
1642
1643
1644
1644
1645
1645
1646
1646
1647
1647
1648
1648
1649
1649
1650
1651
1652
1653
1654
1654
1655
1655
1656
1656
1657
1657
1658
1658
1659
1659
1660
1661
1662
1663
1664
1664
1665
1665
1666
1666
1667
1667
1668
1668
1669
1669
1670
1671
1672
1673
1674
1674
1675
1675
1676
1676
1677
1677
1678
1678
1679
1679
1680
1681
1682
1683
1684
1684
1685
1685
1686
1686
1687
1687
1688
1688
1689
1689
1690
1691
1692
1693
1694
1694
1695
1695
1696
1696
1697
1697
1698
1698
1699
1699
1700
1701
1702
1703
1704
1704
1705
1705
1706
1706
1707
1707
1708
1708
1709
1709
1710
1711
1712
1713
1714
1714
1715
1715
1716
1716
1717
1717
1718
1718
1719
1719
1720
1721
1722
1723
1724
1724
1725
1725
1726
1726
1727
1727
1728
1728
1729
1729
1730
1731
1732
1733
1734
1734
1735
1735
1736
1736
1737
1737
1738
1738
1739
1739
1740
1741
1742
1743
1744
1744
1745
1745
1746
1746
1747
1747
1748
1748
1749
1749
1750
1751
1752
1753
1754
1754
1755
1755
1756
1756
1757
1757
1758
1758
1759
1759
1760
1761
1762
1763
1764
1764
1765
1765
1766
1766
1767
1767
1768
1768
1769
1769
1770
1771
1772
1773
1774
1774
1775
1775
1776
1776
1777
1777
1778
1778
1779
1779
1780
1781
1782
1783
1784
1784
1785
1785
1786
1786
1787
1787
1788
1788
1789
1789
1790
1791
1792
1793
1794
1794
1795
1795
1796
1796
1797
1797
1798
1798
1799
1799
1800
1801
1802
1803
1804
1804
1805
1805
1806
1806
1807
1807
1808
1808
1809
1809
1810
1811
1812
1813
1814
1814
1815
1815
1816
1816
1817
1817
1818
1818
1819
1819
1820
1821
1822
1823
1824
1824
1825
1825
1826
1826
1827
1827
1828
1828
1829
1829
1830
1831
1832
1833
1834
1834
1835
1835
1836
1836
1837
1837
1838
1838
1839
1839
1840
1841
1842
1843
1844
1844
1845
1845
1846
1846
1847
1847
1848
1848
1849
1849
1850
1851
1852
1853
1854
1854
1855
1855
1856
1856
1857
1857
1858
1858
1859
1859
1860
1861
1862
1863
1864
1864
1865
1865
1866
1866
1867
1867
1868
1868
1869
1869
1870
1871
1872
1873
1874
1874
1875
1875
1876
1876
1877
1877
1878
1878
1879
1879
1880
1881
1882
1883
1884
1884
1885
1885
1886
1886
1887
1887
1888
1888
1889
1889
1890
1891
1892
1893
1894
1894
1895
1895
1896
1896
1897
1897
1898
1898
1899
1899
1900
1901
1902
1903
1904
1904
1905
1905
1906
1906
1907
1907
1908
1908
1909
1909
1910
1911
1912
1913
1914
1914
1915
1915
1916
1916
1917
1917
1918
1918
1919
1919
1920
1921
1922
1923
1924
1924
1925
1925
1926
1926
1927
1927
1928
1928
1929
1929
1930
1931
1932
1933
1934
1934
1935
1935
1936
1936
1937
1937
1938
1938
1939
1939
1940
1941
1942
1943
1944
1944
1945
1945
1946
1946
1947
1947
1948
1948
1949
1949
1950
1951
1952
1953
1954
1954
1955
1955
1956
1956
1957
1957
1958
1958
1959
1959
1960
1961
1962
1963
1964
1964
1965
1965
1966
1966
1967
1967
1968
1968
1969
1969
1970
1971
1972
1973
1974
1974
1975
1975
1976
1976
19
```

Memories

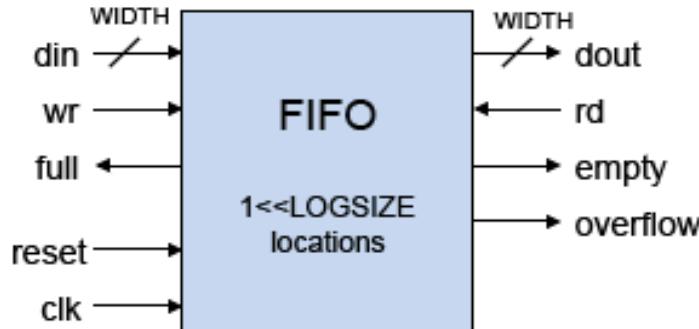
Memories in Verilog

- `reg bit; // a single register`
- `reg [31:0] word; // a 32-bit register`
- `reg [31:0] array[15:0]; // 16 32-bit regs`


```
// combinational (asynch) read
assign read_data = array[index];

// clocked (synchronous) write
always @(posedge clock)
    array[index] <= write_data;
```

FIFOs



```
// a simple synchronous FIFO (first-in first-out) buffer
// Parameters:
//   LOGSIZE (parameter) FIFO has 1<<LOGSIZE elements
//   WIDTH   (parameter) each element has WIDTH bits
// Ports:
//   clk      (input) all actions triggered on rising edge
//   reset    (input) synchronously empties fifo
//   din      (input, WIDTH bits) data to be stored
//   wr       (input) when asserted, store new data
//   full     (output) asserted when FIFO is full
//   dout     (output, WIDTH bits) data read from FIFO
//   rd       (input) when asserted, removes first element
//   empty    (output) asserted when fifo is empty
//   overflow (output) asserted when WR but no room, cleared on next RD
module fifo #(parameter LOGSIZE = 2, // default size is 4 elements
              WIDTH = 4) // default width is 4 bits
  (input clk,reset,wr,rd, input [WIDTH-1:0] din,
   output full,empty,overflow, output [WIDTH-1:0] dout);
  ...
endmodule
```

And finally... Microprocessors



Lots of tools and tutorials...

e.g. NIOS IDE (Integrated Development Environment)
DE2 demonstrations

Tools – SOPC builder

<http://www.altera.com/education/demonstrations/sopc-builder/sopc-builder-demo.html>

ftp://ftp.altera.com/up/pub/Tutorials/DE2/Computer_Organization/tut_sopc_introduction_verilog.pdf

The SOPC Builder is a tool used in conjunction with the Quartus II CAD software. It allows the user to easily create a system based on the Nios II processor, by simply selecting the desired functional units and specifying their parameters.

There are other choices of µ-processors to implements. E.g.: micro
Operating systems can be used. E.g. µ-Clinux™



The lab exercise

cronograph

Quartus II - D:/save/Ambiente de trabalho/PLCD - Projeto e Controlo em Lógica Digital/Lab1_F-CRLab06/DE2_TOP - DE2_TOP.v

File Edit View Project Assignments Processing Tools Window Help

Project Navigator -

Entry	Logic Cells	Dedicated Logic Reg
Cyclone II EP2C35F672C6	134 (134)	102 (102)

Hierarchy Files Design Units

Tasks Flow Compilation

Task	Time
Compile Design	00:00:38
Analysis & Synthesis	00:00:10
Filter (Place & Route)	00:00:21
Assembler (Generate programming files)	00:00:04
Classic Timing Analysis	00:00:03
EDA Netlist Writer	
Program Device (Open Programmed)	

Message

Type Message

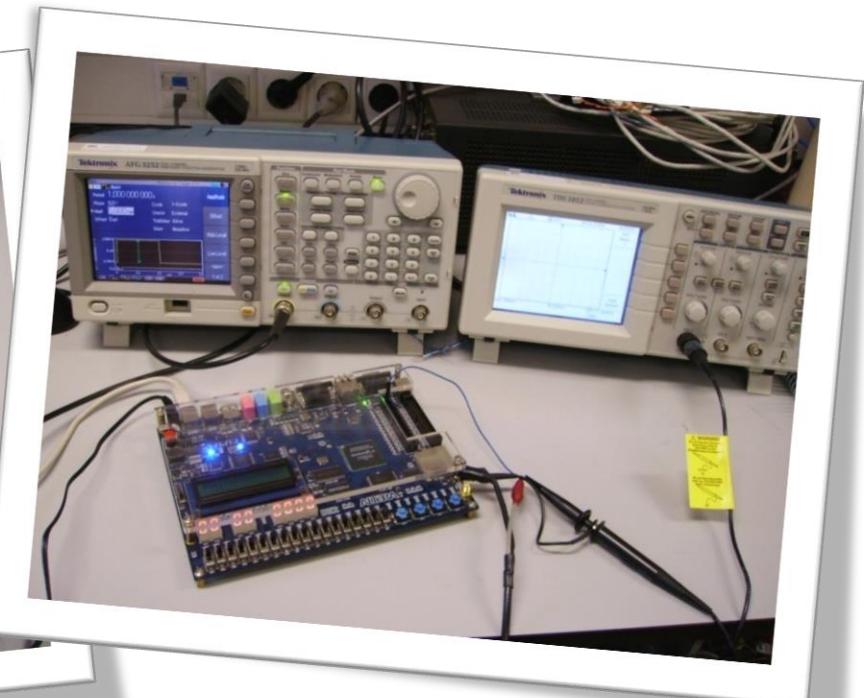
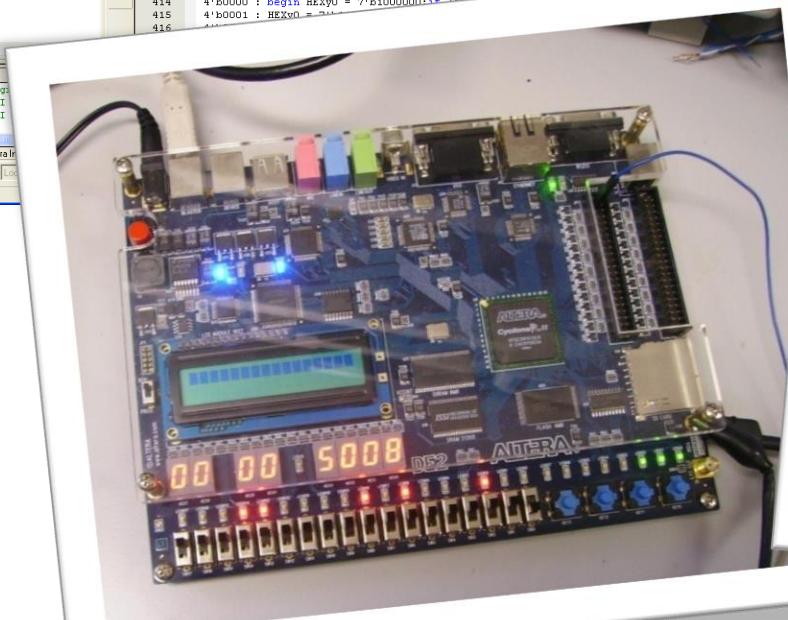
- Info: th for reg
- Info: Quartus II
- Info: Quartus II

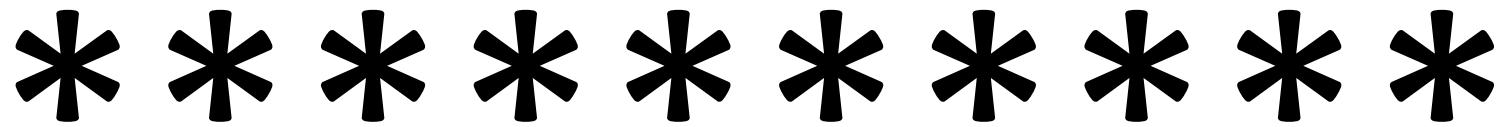
For Help, press F1

Message: 0 of 1507

Code Block (DE2_TOP.v):

```
385     else
386         HEXy00=7'b1000000;
387
388         Cont <= Cont+1;
389         if (Cont>24998)
390             begin
391                 Cont [14:0] = 0;
392                 millis = !millis;
393             end
394         end
395     end
396
397     assign LEDR[14:0]=Cont[14:0];
398     assign LEDG[0]=millis;
399
400 //----- milli -----
401
402     always0 (posedge millis or negedge KEY[3])
403     begin
404         if (KEY[3]==0)
405             begin
406                 Cont_milli<=1;
407                 HEXy0=7'b1000000;
408             end
409         else
410             begin
411                 Cont_milli <= Cont_milli+1;
412                 case (Cont_milli)
413                     4'b0000 : begin HEXy0 = 7'b1000000;end
414                     4'b0001 : HEXy0 = 7'b00000001;end
415                     4'b0010 : HEXy0 = 7'b00000000;end
416             end
417     end
418 
```



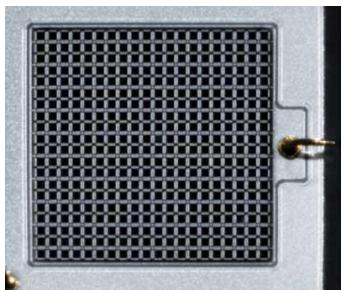
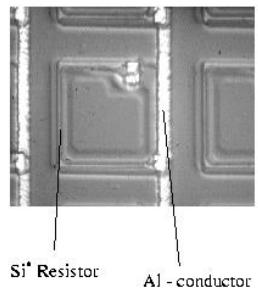


SiPMs ...

SiPM – Silicon PhotoMultiplier



Foto-Diodo de avalanche
em modo Geiger com
resistência Quenching Resistor



SiPM pixel



SiPM Matrix

SiPMs – CMOS binário
1 célula tem sempre o mesmo sinal
Saída: Soma de várias células
Sinais “digitalizados”
Ideais para “Single Photon Counting”
Eficientes
Podem ser expostos a luz
Tensões baixas (<100v)

Problemas:

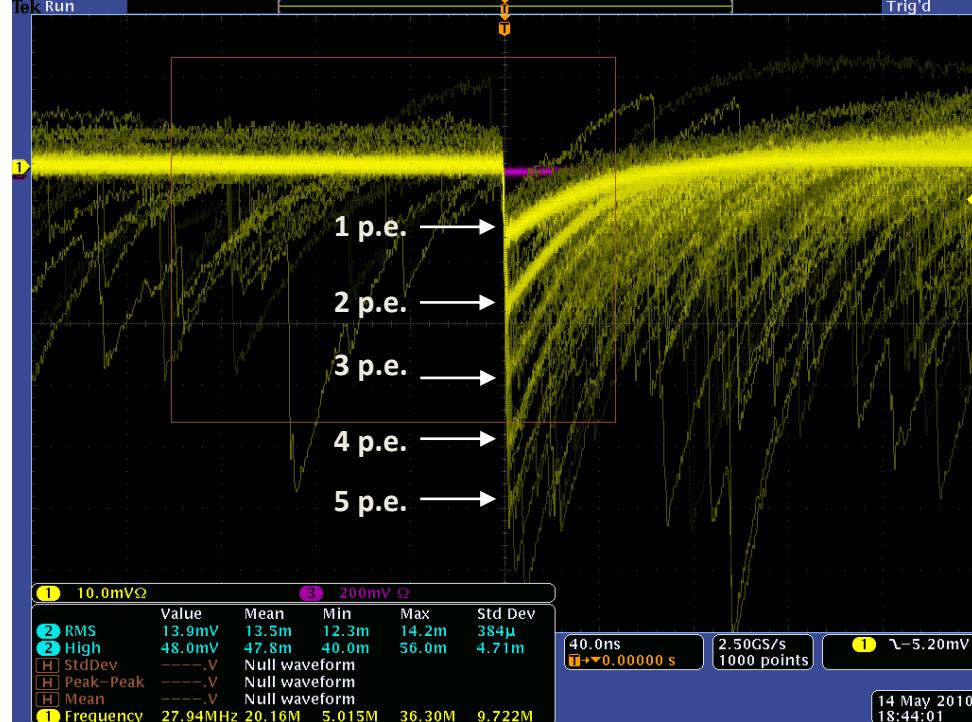
Dependência com a temperatura;
Dependência com a tensão
Ruído
Crosstalk

No LIP...

Caracterização dos SiPM

Sistema controlo temperatura ($\sim -20^{\circ}\text{C}$)

Sistema leitura 64 canais



Substituir uma câmara de Auger:

Auger: $800 \text{ mm} \times 800 \text{ mm} = 6.4 \times 10^5 \text{ mm}^2$

1 SiPM: $3 \text{ mm} \times 3 \text{ mm} = 9 \text{ mm}^2$

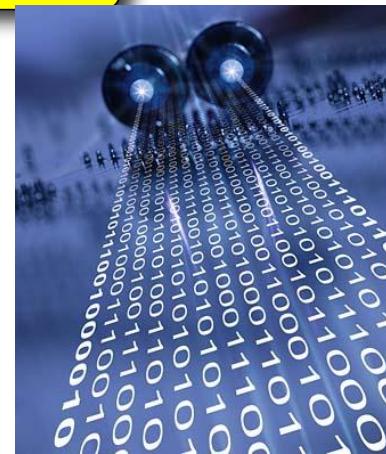
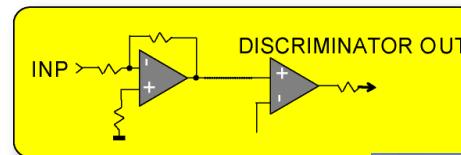
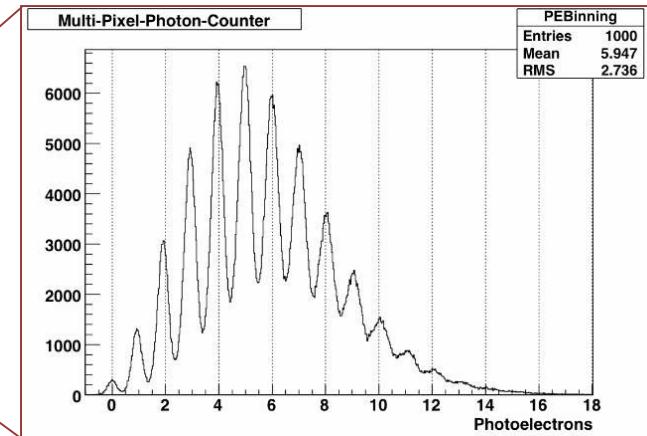
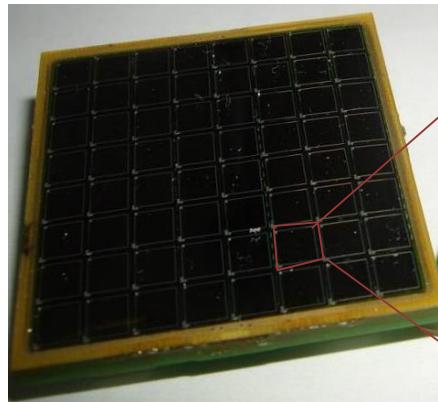
Nº canais da ordem de 7×10^4

Ganhos:

Resolução

Eficiência do detector

Photon counting

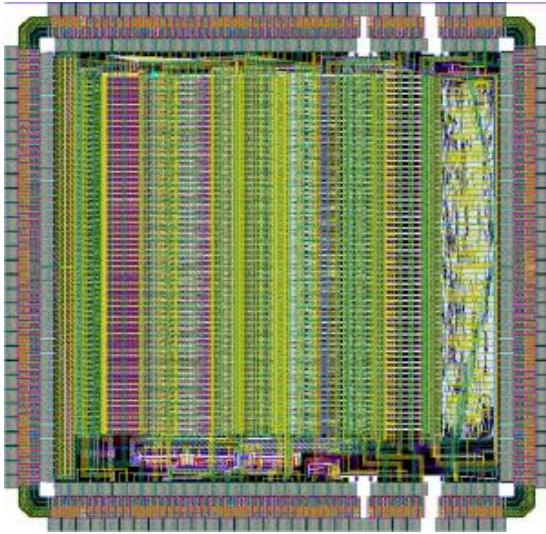


Threshold simples: Digital

Vários Thresholds: Aumentar gama dinâmica

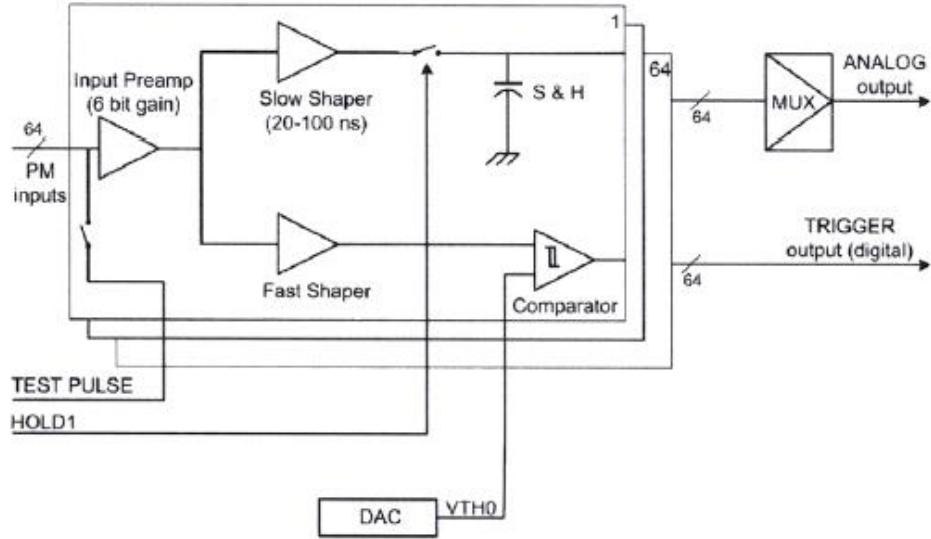
Pode ser implementado num ASIC: Muitos Canais

O ASIC de front-end



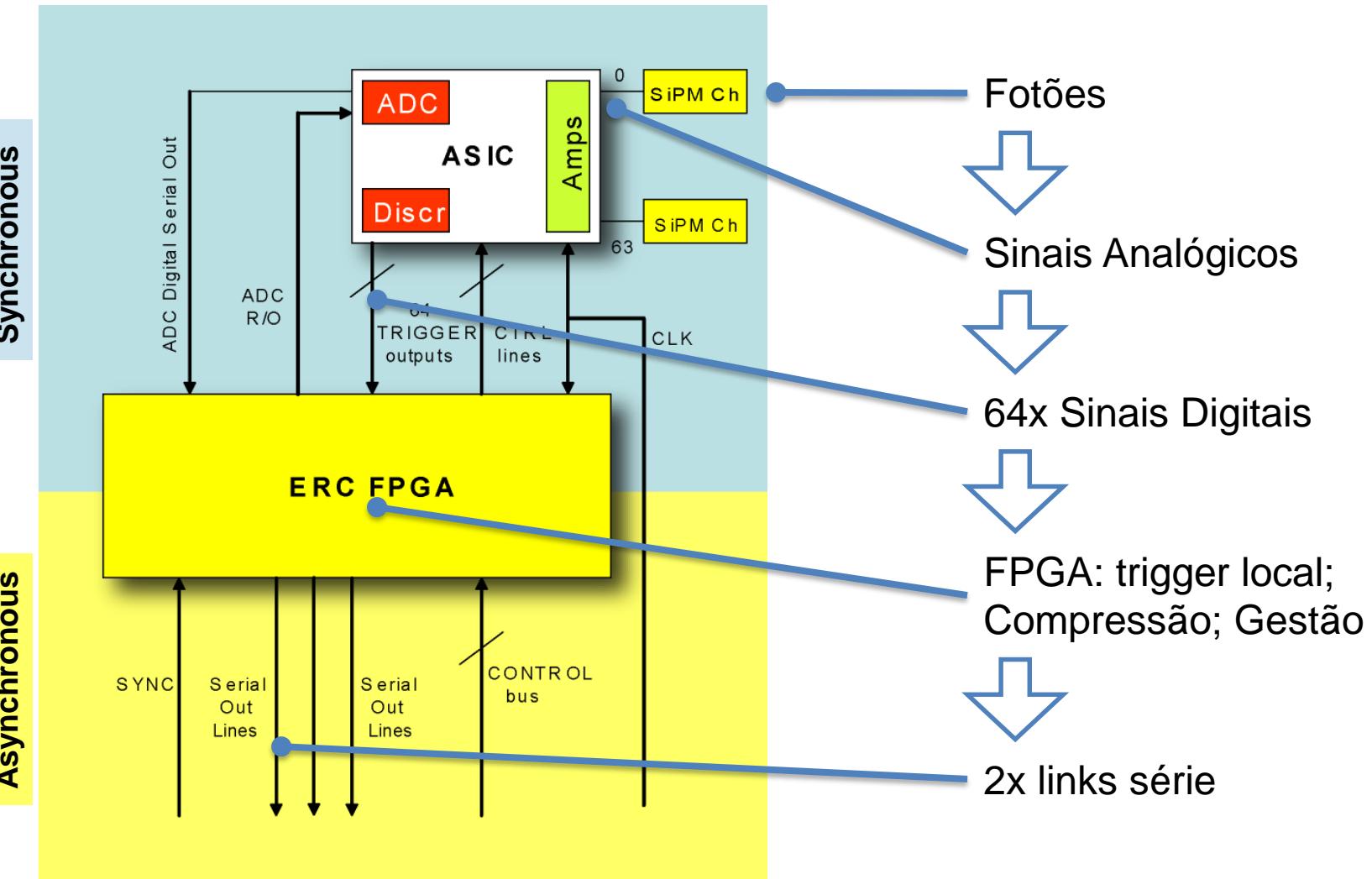
Baseline Option
ASIC MAROC3

Omega

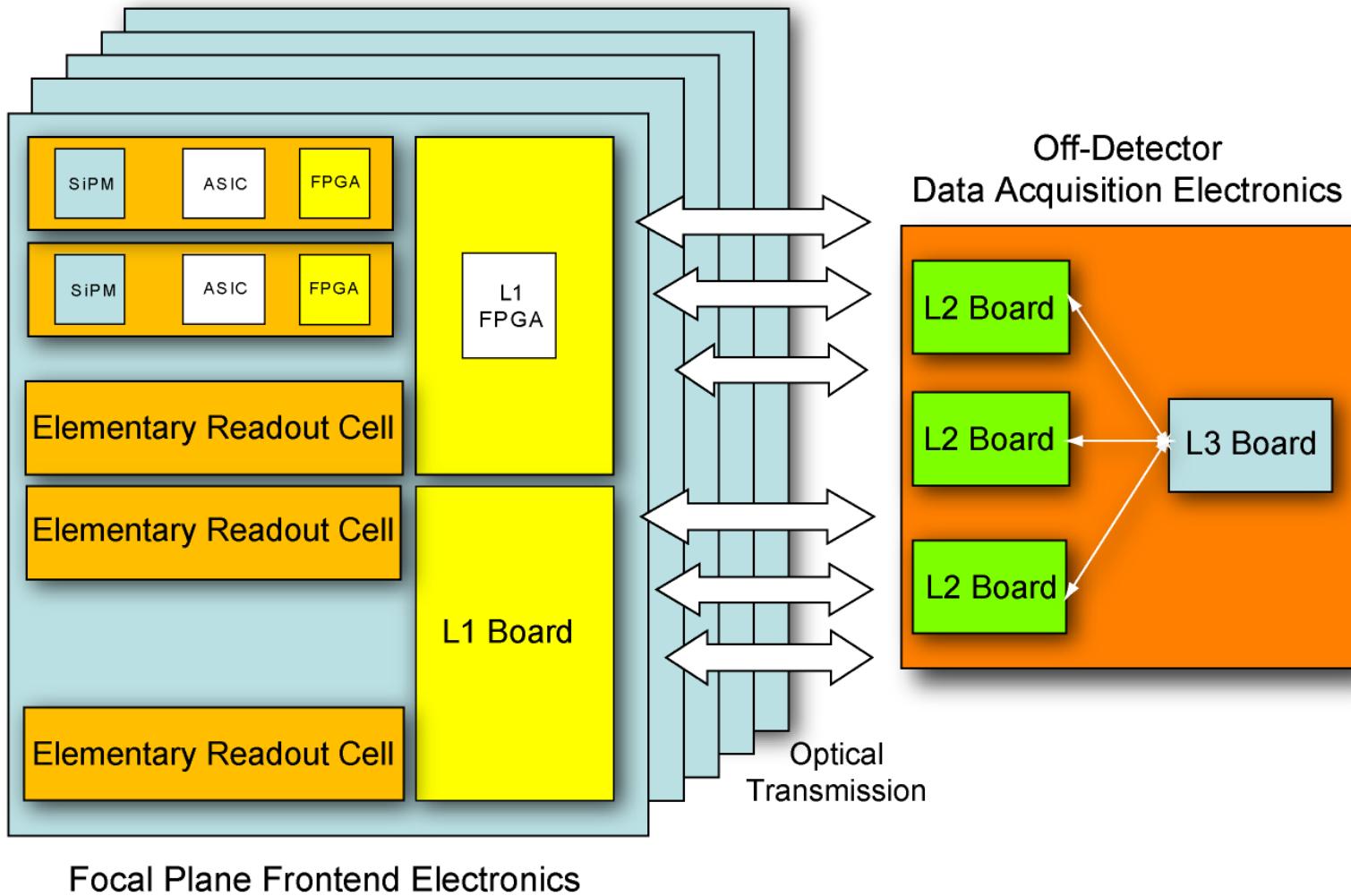


64 pré-amplificadores
Ganho programável por canal
64 sinais digitais de saída
ADC 12 bits

ERC - Elementary Readout Cell



Aquisição de Dados em pirâmide...



L2/L3 boards

Solução Comercial:

Placas MicroTCA (industria Telecomunicações)

Vários links ópticos por placa

Podem ser instalados numa crate com bus

FPGAs Rápidas

Reprogramáveis

Scalable solution



PC Optical Link Board



Backplane

Optical Links

Altera® Stratix® IV GX FPGA
16 x 6.25 GHz SerDes transceivers
Optical Links (3.125 Gbps) or SFP+ (6.25 Gbps)
2 x 1 GB SDRAM

As Placas podem ser
programadas como placas de
trigger de nível L2 ou L3

Hardware

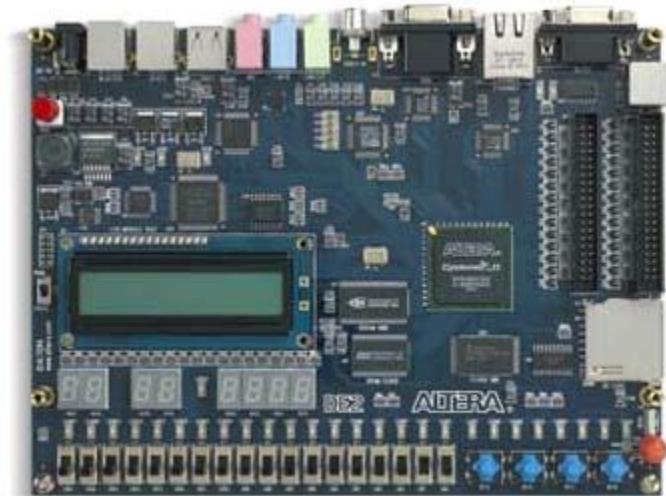
<http://www.altera.com>

<http://www.terasic.com.tw/en/>

Development boards from ~\$100



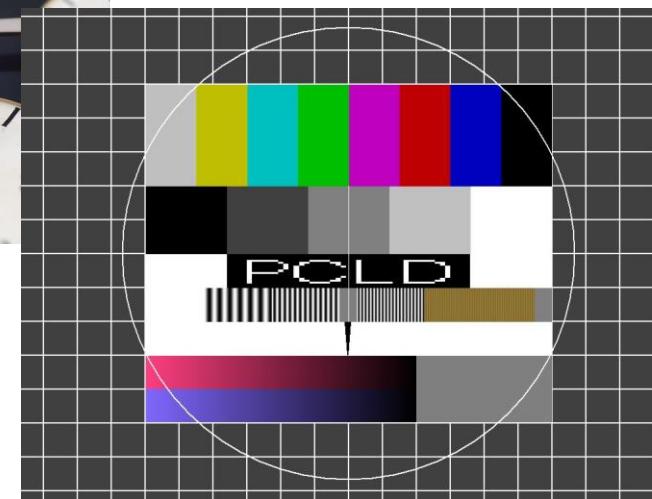
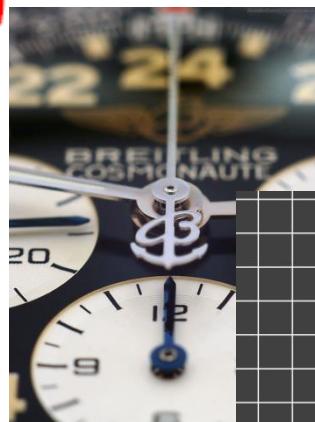
3-4 semanas aulas
14 semanas laboratório



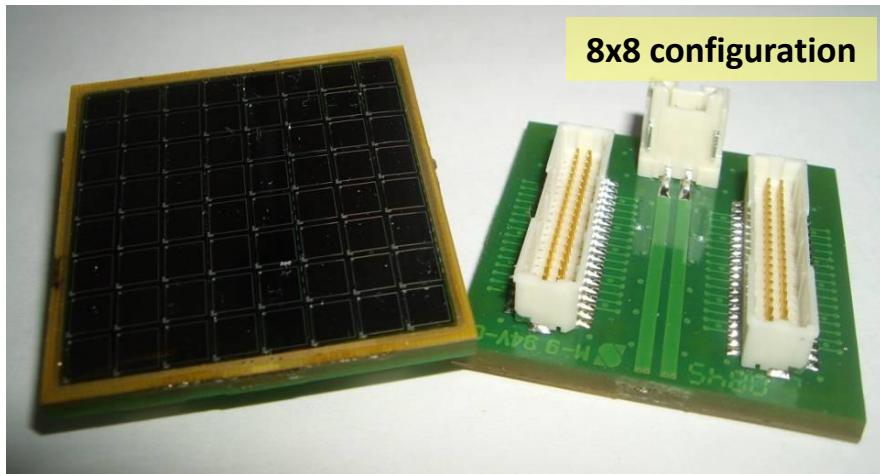
4 exercícios:
•“Olá Mundo”
•“Breitling”
•Mira Técnica
•Jogo

1 Trabalho final “negociado”
Caderno de encargos define requisitos

Olá Mundo



SiPM Multi-Pixel Arrays



Zecotek Photonics, MAPD3-N device

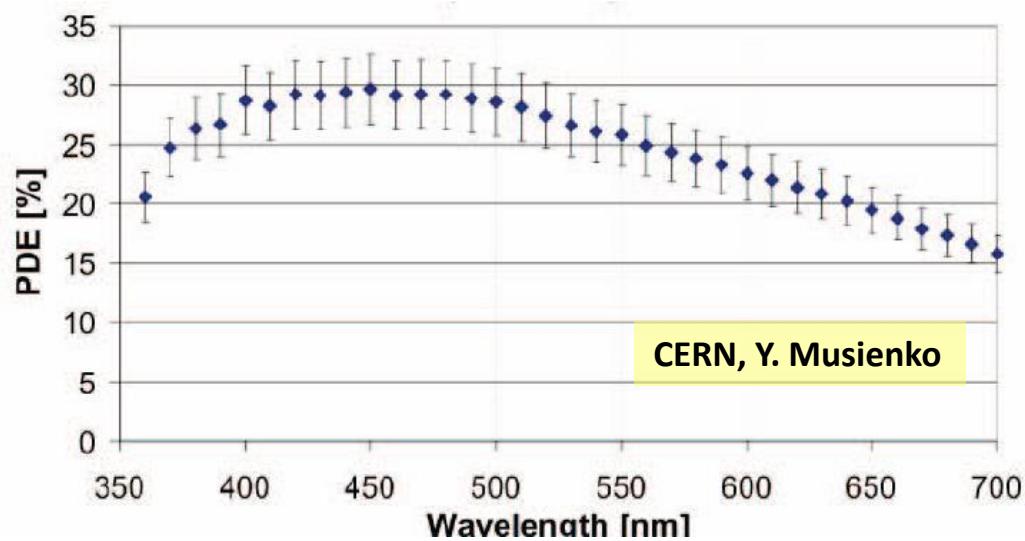
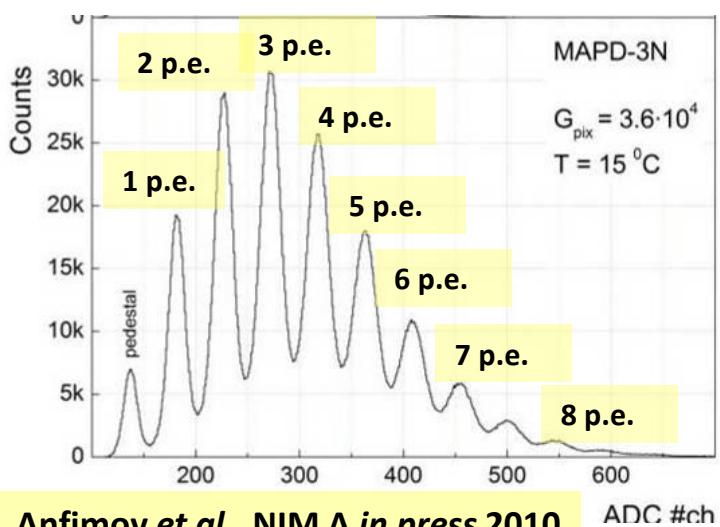
Gain: $7 \times 10^4 - 1 \times 10^5$ (11-15 fC for 1 p.e.)

Dark current: 1-3 MHz per $3 \times 3 \text{ mm}^2$ pixel

Temperature sensitivity (dM/dV) $\sim 5\%/\text{°C}$

Crosstalk: 5-15%

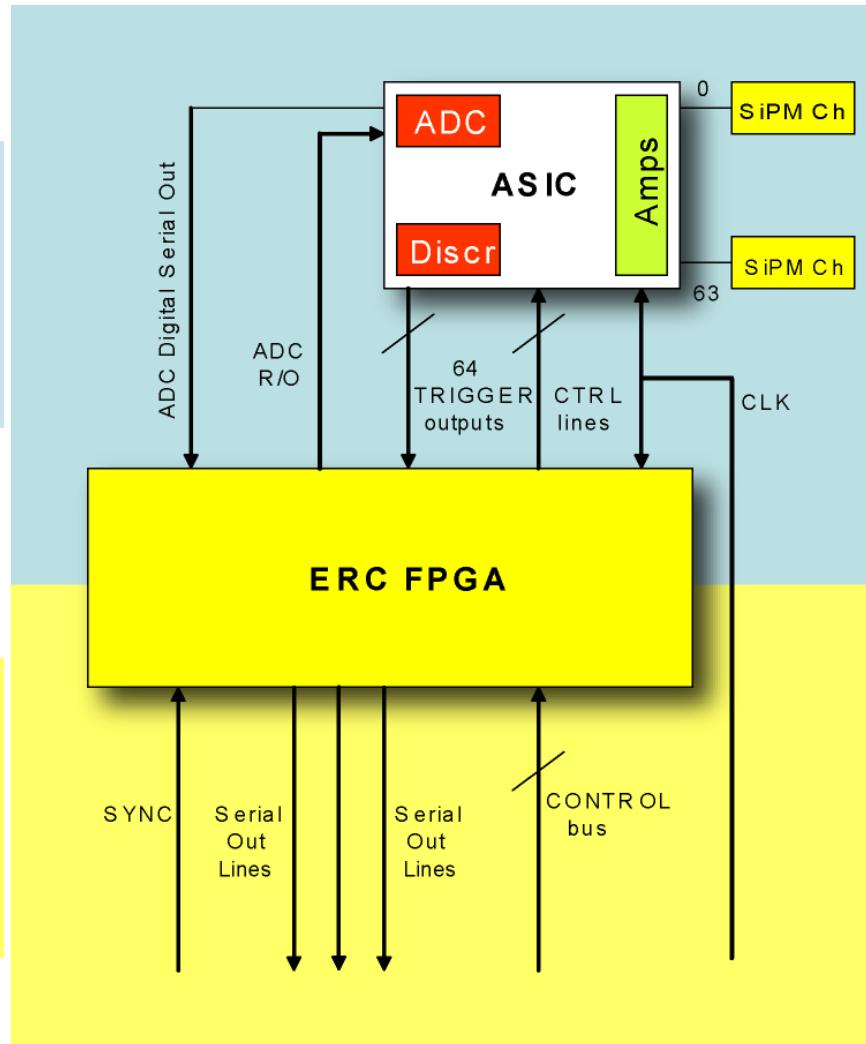
Bias supply: 90 V, common cathode



Anfimov *et al.*, NIM A *in press* 2010

ERC - Elementary Readout Cell

Synchronous



Asynchronous

64 SiPM (8x8 configuration) array

64 input channel ASIC with 64 discriminators

Charge output available as option

Local clock at 40/80 MHz

Total bandwidth between ASIC and ERC FPGA: 2.56 Gbps at 40 MHz or 6.4 Gbps at 100 MHz

ERC FPGA performs zero suppression and local threshold

Output serial links (LVDS, min 100 MHz)

Typical payload: 64 bits + 40 bit Timestamp (clock counter) + overhead + trailer ~ 110-120b

2x100 Mbps LVDS link compatible with ~2 MHz event rate / 64 channels)

Main options

Signals Digitized and time-tagged “as soon as possible”.

Data forward @ max bandwidth using off-the-shelf links and boards.

- A clock is distributed only to front-end boards, for time marker assignment
- Each trigger primitive has a time marker assigned by the front-end board
- All trigger primitives from front-end boards sent asynchronously to trigger boards via data links running at their own clocks, for maximum bandwidth
- In an asynchronous system, each trigger level works at its own clock frequency
- No timing procedures are required

From Industry: Fast Data transfer standards. E.g.

- xTCA from Telecom Industry
- Inherent asynchronous (network switching)
- Data and slow-control data streams can be send over the same data lines, reducing the number of cables required

Bonus: DAQ architecture can re-use standard MA-PMTs