

Duke

Progress Report II

Team Members:

Xuan Bao Kai Wang Xiaoyan Yin Yang Jiang

ECE 261

Project 2008





Index







Duke

Overview of Our Present Step



This is the top level schematic of our project.

Every block here has already been fully defined.

All

components and the whole system have correctly passed digital simulation.





Introduction to the FIFO



This is the schematic of one FIFO component. It contains 8 Dflipflops with reset (for further testing) and works as a buffer for both input and output.

www.themegallery.com

Company Logo





Duke

Introduction to the FIFO

• :FIF0:B0 2 • <td< th=""><th></th></td<>	
<pre>** :FF0:A 4</pre>	
<pre>:** :FIF0:8 2 DC(s)7/s /s/4/02 U(D)COC(A2C(s)COC) S (WaCO(2)s) / (D)COC(COC) S /2/s / (S)/32 (D)COC(COC) / (D</pre>	
• :FIFO:CLK -1	
<pre> FIFOCLK_INV O FIFO.Reset_INV -1 -1 -1 -1</pre>	
<pre> FFF0:Reset_INV -1 'celldefine 'suppress_faults 'enable_portfaults 'timescale lns/10ps module dflipflop (D,CLK,CLK_in output Q; input D,CLK,CLK_inv,Reset_i reg Q; reg Q; </pre>	
<pre>`suppress_faults `enable_portfaults `timescale 1ns/10ps module dflipflop (D,CLK,CLK_irr output Q; input D,CLK,CLK_inv,Reset_i reg Q;</pre>	
<pre>`enable_portfaults `timescale 1ns/10ps module dflipflop (D,CLK,CLK_ir output Q; input D,CLK,CLK_inv,Reset_i reg Q;</pre>	
<pre>`timescale lns/10ps module dflipflop (D,CLK,CLK_ir output Q; input D,CLK,CLK_inv,Reset_i reg Q;</pre>	
module dflipflop (D,CLK,CLK_ir output Q; input D,CLK,CLK_inv,Reset_i reg Q;	
input D, CLK, CLK_inv, Reset_i reg Q;	<pre>/,Reset_inv,Q);</pre>
reg Q;	av;
always @ (posedge CLK)	
if (~Reset_inv) begin	
$Q \leq 0;$	
end else begin	
$Q \leq D;$	

This is the digital simulation result of FIFO component. All possible patterns have been exhaustively tested. To make top level testing easier, we then rewrote the verilog for Dflipflop. The key part of codes is shown above.





Index







Half Adder





www.themegallery.com

Company Logo

Duke





Full Adder

MCC Ele Est Act Seint, Entre Montherene Encort Leve Heb Act 28 월 전 프 속 역, 및 단가 문 문 편 Act 6+ (V)(abs)(Hal,Acier Johannik (Jahent)())()	(5.3963, 3.7027) Hotkeys: On		Input		Out	tput
Schematic.#2: Big_Project / Ful_Adder / Schematic	Schematic edit	Ain	Bin	Cin	Sum	Cout
XOR22: VIEVs Barauto	Draw Text Check & Save Select	0	0	0	0	0
M_logic M_logic M_logic M_logic M_logic M_logic M_logic M_logic M_logic M_logic	By Property Unselect All Eder Move	0	0	1	1	0
TI X0R23 VIEve Blor Fault CO VIEve Blor Fault CO	Copy Delete Undo Filp •	0	1	0	1	0
NL.logic NL.logic NL.logic NL.logic VI.logic TOP. A.D0. decesso TOP. A.D0. decesso TOP. A.D0. decesso VI.logic TOP. A.D0. decesso TOP. E.D0. decesso TOP. E.D0. decesso VI.logic TOP. A.D0. decesso TOP. E.D0. decesso TOP. E.D0. decesso	Rotate Properties	0	1	1	0	1
AND 22 VIEW- Storf aut CO VIEW- Storf aut CO	Bus/Bundle Add Source Property Near	1	0	0	1	0
A Number 2000 Tops 1, 0, 0, 000000 Tops 1, 0, 0, 000000 Tops 1, 0, 0, 000000 Tops 1, 0, 0, 0, 000000 Tops 1, 0, 0, 0, 000000 Tops 1, 0, 0, 0, 0, 000000 Tops 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,	Instance Net Miscellances Connect	1	o	1	0	1
		1	1	0	0	1
Alexa eff mouse helton to complete move. Alexas eff mouse helton to complete move.		1	1	1	1	1

~

C2



wave - default

www.themegallery.com

Company Logo

- - -

ď

Verilog for Half Adder and Full Adder:

```
celldefine
suppress faults
enable_portfaults
'timescale 1ns/10ps
module halfadder (Co,S,A,B);
 output S, Co;
  input A,B;
  assign {Co,S}= A+B;
   parameter TDR_A_Co = 0.000;
   parameter TDF_A_Co = 0.000;
    parameter TDR_B_Co = 0.000;
   parameter TDF_B_Co = 0.000;
   parameter TDR_A_S = 0.000;
   parameter TDF_A_S = 0.000;
   parameter TDR B S = 0.000;
   parameter TDF_B_S = 0.000;
   'ifdef functional
     initial $display("\n *** ATTENTION! Functional Simulation,
   else
     ifdef TETRAMAX
   else
   specify
  'ifdef INCA
     showcancelled S;
     pulsestyle_ondetect S;
  else
     `ifdef verilog
        $showcancelled;
     $pulsestyle ondetect;
      endif
  endif
      (A \Rightarrow Co) = (TDR A Co, TDF A Co);
      (B => Co) = (TDR B_Co, TDF B_Co);
      (A => S) = (TDR_A_S, TDF_A_S);
      (B => S) = (TDR_B_S, TDF_B_S);
   endspecify
   'endif
   endif
endmodule
disable_portfaults
'nosuppress faults
 endcelldefine
```

```
celldefine
suppress_faults
enable_portfaults
'timescale 1ns/10ps
module fulladder(Cout, Sum, Ain, Bin, Cin);
input Ain, Bin, Cin;
output Sum, Cout;
 assign {Cout, Sum}=Ain+Bin+Cin;
   parameter TDR_Ain_Cout = 0.000;
   parameter TDF_Ain_Cout = 0.000;
   parameter TDR Bin Cout = 0.000;
   parameter TDF_Bin_Cout = 0.000;
   parameter TDR_Cin_Cout = 0.000;
   parameter TDF_Cin_Cout = 0.000;
   parameter TDR_Ain_Sum = 0.000;
   parameter TDF_Ain_Sum = 0.000;
   parameter TDR_Bin_Sum = 0.000;
   parameter TDF_Bin_Sum = 0.000;
   parameter TDR_Cin_Sum = 0.000;
   parameter TDF_Cin_Sum = 0.000;
'ifdef functional
     initial $display("\n *** ATTENTION! Functional Simulation
   else
     ifdef TETRAMAX
   'else
  specify
  'ifdef INCA
     showcancelled Sum:
    pulsestyle_ondetect Sum;
  else
     'ifdef verilog
        $showcancelled;
     $pulsestyle_ondetect;
     endif
  endif
      (Ain => Cout) = (TDR_Ain_Cout, TDF_Ain_Cout);
      (Bin => Cout) = (TDR_Bin_Cout, TDF_Bin_Cout);
      (Cin => Cout) = (TDR_Cin_Cout, TDF_Cin_Cout);
      (Ain => Sum) = (TDR_Ain_Sum, TDF_Ain_Sum);
      (Bin => Sum) = (TDR_Bin_Sum, TDF_Bin_Sum);
      (Cin => Sum) = (TDR Cin Sum, TDF Cin Sum);
  endspecify
   endif
   endif
endmodule
disable_portfaults
nosuppress faults
```

endcelldefine







8-bit Ripple Carry Adder



wave - default											
	13	64	-14			18	26	-6		-30	13
🖽 🔶 :carryadder:B	-29	16	-24	-17	15				-17	-29	
.carryadder:Z	-16	80	-38	-31	1	33	41	1 9	-23	-59	-16
• carryadder:Z	-16	80	-38	-31	1	33	41	<u>,</u> 9	-23	-59	-16







8-bit Ripple Carry Subtractor



🗇 :carrysub:Cin	-1											
🗉 🔶 :carrysub:A	31	-19		-3		-15		29		31		
🖃 🔷 :carrysub:B	-7	21	-11		-13		-29		-31	25	-7	
.carrysub:Z	38	-40	-8	8	10	-2	14	58	60	6	38	





Index





Unsigned Multiplication:



Signed Multiplication:

4*4 Bits Signed Multiplier

Quick Overview...

a_1 a_{\star} a_3 a_2 a_1 a_0 as a_3 a_2 a_0 X_4 Хς x_2 x_1 xo X_{4} X_2 x_2 x_1 X_0 $a_3 x_0$ $a_2 x_0$ $a_1 x_0$ $a_4 x_0$ $a_0 x_0$ $a_2 x_0$ $a_1 x_0$ $a_4 x_0$ $a_3 x_0$ $a_0 x_0$ $a_2 x_1$ $a_1 x_1$ $a_4 x_1$ $a_3 x_1$ $a_0 x_1$ $a_2 x_1$ $a_1 x_1$ $a_0 x_1$ $a_4 x_1$ $a_3 x_1$ $a_1 x_2$ $a_2 x_2$ $a_3 x_2$ $a_2 x_2$ $a_0 x_2$ $a_{\star}x_{2}$ $a_3 x_2$ $a_1 x_2$ $a_0 x_2$ $a_4 x_2$ $a_{2}x_{3}$ $a_1 x_3$ $a_0 x_3$ $a_4 x_3$ $a_3 x_3$ $a_2 x_3$ $a_1 x_3$ $a_4 x_2$ $a_2 x_2$ $a_0 x_3$ $a_3 x_4$ $a_2 x_4$ $a_1 x_4$ $a_{4}X_{4}$ $a_0 x_4$ $a_4 x_4$ $a_3 x_4$ $a_2 x_4 a_1 x_4$ $a_0 x_4$ 1 p_{0} p_8 p_7 p_6 p_5 p_4 p_3 p_2 p_1 p_0 p_0 p_8 p_6 p_5 p_4 p_3 p_2 p_1 p_7 \mathcal{P}_0

Algorithm: shift and addition

Algorithm: Baugh-Woodley technique

Substitute AND2 gates to NAND2 gates, plus inverter... Schematics?

www.themegallery.com

Company Logo

719-01 11 shéi 💽 H.HELL 龖 쀠翻 雔 ٢ PLATE AND DO a contraction . Had at the Lo 4 χ ULL ARRESS FULLARIES FILLING ST and posterior in th 11 12 PLAN AREA 7 FILL COMPLETE ALL ADDRESS ALL PARTY OF - 13 2 INV2 9.999999 9.999999

Schematic#2 Big Project / Multiplier / Schematic





Verification of Multiplier's Results:

4	:Multiplier:VDD	-1																
	:Multiplier:X	0	0	-3		-4	-8	1	4		-3	-7	-8		5		4	0
•	:Multiplier:Y	-8	-8	-5 3	2		1	-7 -8	-5	2		1	0	-8	-5	-6	2	1
•	:Multiplier:Z	0	0	15 -9	-6	-8	-8	-7)-8	-20	8	-6	-7	-8 (0	64	-25	-30)8	0

Results: 0*8=0 -3*-5=15 -3*3=-9 3*2=-6 -4*2=-8

Realizes the multiplication of input X and Y, with the correct outcome Z=XY, which will be sent to Ripple Carry Adder and Subtractor to calculate: AD+BC & AC-BD

Consists of:

Nand2	× 6
And2	× 10
Full Adders	× 9
Half Adders	× 3
Inv	× 1





Index









System schematic









System Simulation





www.themegallery.com







Feature estimation

Number of transistors

	Total			
SUM*1	DIFF*1	MULT*4	FIFO*6	transistors
252	288	434*4	256*6	3812

***Area (mm^2)**

	120%*Total			
SUM*1	DIFF*1	MULT*4	FIFO*6	area
0.028	0.033	0.047*4	0.030*6	0.515







Feature estimation

Power consumption

$$P = \alpha \cdot C \cdot V^2 \cdot f$$

= 0.1 \cdot [3812 \cdot (12\lambda) \cdot (0.8\mum/2\lambda) \cdot (2fF / \mum)] \cdot 5^2 \cdot 10^6
= 0.0915mW / MHz







Floor Plan



www.themega

Company Logo







To be continued...



Company Logo