4 A brief account of any design choices you made and reasons why.

1. Data Buffer: The required buffer is positive edge with active low reset. Hence a D -flip-flop with D input, Q output, edge clock in and low reset is been chose from the library. i.e. DFFSRLQ (Positive-edge triggered D-flipflop with synchronous reset (active low) and Q-output only)



- 2. Control Unit: A moore machine with clocked output is implemented for control unit of the LDI86. It has Eight state with seven different signals.
- 3. Multiplexor: This unit is full custom cell. First, using PMOS and NMOS a 1-bit multiplexor is made. Then this cell is used for 8-bit as an 8-bit multiplexor.

An ASM chart for the control unit.

STATE	NOP	LDA	TBD	SBA	ABT	NOP	TTD	TAB
SIGNAL								
M(2:0)	110	100	011	000	001	110	101	010
ADDSHIFT(1:0)	01	01	01	10	01	01	01	01
AEN		1	0	0	0		0	0
BEN		0	0	0	0		0	1
TEN		0	0	1	1		0	0
DEN		0	1	0	0		1	0
DATALOAD	0	1	1	1	1	1	1	1





library IEEE; use IEEE.std_logic_1164.all;

entity C_UNIT is port (CLOCK, RESET : in std_logic; AEN,BEN,TEN,DEN,DATALOAD :out std_logic; M: out std_logic_vector(2 downto 0); ADDSHIFT: out std_logic_vector(1 downto 0)); end C UNIT; architecture BEHAVIOUR of C UNIT is

type STATE_TYPE is (ZERO,ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT);

signal STATE: STATE_TYPE; begin FSM : process (CLOCK, RESET) begin if RESET = '0' then STATE <= ZERO; elsif CLOCK'event and CLOCK ='1' then case STATE is when ZERO =>

STATE <= ONE; when ONE => STATE <= TWO; when TWO => STATE <= THREE; when THREE => STATE <= FOUR; when FOUR => STATE <= FIVE; when FIVE => STATE <= SIX; when SIX => STATE <= SEVEN; when SEVEN=> STATE <= EIGHT; when EIGHT => STATE \leq ONE; end case; end if; end process FSM; "110" when (STATE = ONE) else $M \le$ "100" when (STATE = TWO) else "011" when (STATE = THREE) else "001" when (STATE = FIVE) else "110" when (STATE = SIX) else "101" when (STATE = SEVEN) else "010" when (STATE = EIGHT) else "000"; ADDSHIFT <= "10" when (STATE = FOUR) else "00" when (STATE = ZERO) else "01"; AEN<= '1' when (STATE = TWO) else **'**0'; '1' when (STATE = EIGHT) else BEN<= **'0'**: TEN<= '1' when (STATE = FOUR) else '1' when (STATE = FIVE) else **'**0'; '1' when (STATE = THREE) else DEN<= '1' when (STATE = SEVEN) else **'**0': DATALOAD<= '0' when (STATE = ONE) else **'**1'; end BEHAVIOUR;

TESTBENCH FOR FSM CONTROL UNIT

library ieee; Use ieee.std_logic_1164.ALL;

Entity tstbnch_cu is end tstbnch_cu;

architecture test of tstbnch cu is

component CTRL_UNIT

addshift: out std_logic_vector(1 downto 0)); end component; signal reset,clock: std_logic;

signal aen,ben,ten,den,dataload: std_logic; signal m: std_logic_vector(2 downto 0); signal addshift: std_logic_vector(1 downto 0);

begin comp_ut: CTRL_UNIT port map(reset,clock,aen,ben,ten,den,dataload,m,addshift);

--generate a 352.8 kHz clock (8x44.1khz) clock_gen : process begin clock <= '0', '1' after 1400 us; wait for 2800 us; end process;

intialisation:process begin reset <= '0'; wait for 5600 us; reset <= '1';

wait for 22400 us; wait;

end process; end test;

4 A brief analysis of the testability of the design and how this could be improved.

LDI86 is divided into following four major blocks.

- 1. Data Buffer (made by using standard library)
- 2. Processor (IP core)
- 3. Control Unit (VHDL)
- 4. Multiplexor (full custom component)

1. Data Buffer

Schematic of Data buffer using standard library of D-flip flop



Simulation of Data buffer

Undo	Cut		Сору	Paste	Delete	Zoom	αInX	ZoomOu	tΧ	ZmOutXFull	Empland	CrtMark
Tim Sim En	eA = 113.4 d = 170.16	us us			Cursor1 :	= 34.19 C	28135 Cursor	<mark>us</mark> 2 = 56.	72 u:	3	. Time?	= 113.
ur2-Cur1 =	22.591865	us	0			50				100)	
Group: A												
		1	ſ									
		1										
		Ū										
		1										
		Ū.										
		1										
		Ũ										
		1										
		Ū										
		1										
		0										
		1										
		0										
		1										
		0										
		1										
		0										
	Q7 =	1										

2. PROCESSOR SYNTHESIS of IP core



3.Control Unit SYNTHESIS of VHDL code

								<u>s</u> -									
																•	
															بر		
				╞				ŀ	ŀ			ľ	F				гf
												•					
										j.			Ð		Ð		Ð
															·.		
																	•
					į	L					ſ	·					•
					_									Γ			
												•			<u> </u>		
									·	•		•	:		·		
						Ð				Đ-		•	₽		·		
						jt] .			
						-				÷							
						ļ				7.			j.				
															·		
	·																
	•						•	•		-		•	•			•	
										. :							
Ð-		Đ-				0				Ð		•	Ð-			· ¢	Ð
																	•
						Ì											•
	Ī			₽						₩			╟				, ,
	ľ																
													:				
			- Гъ-			Þ				Đ			Ð		-2-		
													1				
			÷D														
			-i _														
						•							-				
						·									•	•	•

VHDL Simulation of control unit





4. Multiplexor 2 to1

Test Schematic of 2 -1 full custom multiplexor cell:



Simulation waveform:



Layout and abstract of 2-1 full custom multiplexor cell:



Verilog functional model:

module mux21 (F, A, B, Select);
 output F;
 input A;
 input B;
 input Select;
reg F;
always @(select or A or B)
F=Select ? B : A;
endmodule

Verilog Simulation of 2-1 multiplexor

ido Cut	Сору	Paste	Delete	ZoomInX	ZoomOutX	ZmOutXFull	Empland	CrtMarke	er DesBrows:1	Bina
TimeA = 0(0) s Sim End = 560 s	TimeA :	= 0(0) s			Cursor	2 = 244 s			Cursor1 = 447	8
Cur2-Cur1 = -203 s	0	- 50 	100 <u>19</u>	50 ₁ 200	,250	,300	<u>,</u> 350 j	400	450 ,500	
select = 1										
a = 1 b = 0]					_		
0 = 0 £ = 0]	ſ						

5. Multiplexor 8 to1 (combination of 2 -1 multiplexor to form 8-1 mux) Schematic



Verilog Simulation of 8-1 multiplexor

ndo	Cut		Сору	Paste	Delete	ZoomInX	ZoomOutX	ZmOutXFull	Empland	CrtMarker	DesBrows:1	- 83
Tin Sim End	meA = 0(0)) = 500,000)	ps ps	Cursor2	Cu: = 0 ps : = 0(0) ps	rsor1 = 76,9	977 ps						
2-curl p: Ainp	= -16,911] ut	ps	0		,100, 000		,200, 000	3	00,000	400	, 000	
A%[B%[0:7] = 'h 5 0:7] = 'h A	5 A	55 AA) (
F%[0:7] = 'h A	A	55	AA	55	AA	55 .	AA 5	5 2	A 55	<u>AA</u>	



Chip: Synthesised Schematic of chip LDI86:

Placement/routing of chip LDI86:



Post layout simulation

Undo Cut Co	ppy Parte	Delete	ZoomInX	ZoomOutX Z	mOutXFull E	Expand CrtM	arker DesBrow	is ;1 Bi
TimeA = 0(0) ps im End = 191,800,000 ps	Cursor1 = 1 Cursor2 = 1 TimeA = 0(0) ps	.3, 748, 375 ps 14, 000, 000 ps						
Cur2-Cur1 = 251,625 ps 0		50, 0	00, 000		100, 000, 000		150,000,000	
Group: A								
CLK = 1			MMM			תתתתת		WWW
RESET = 1								
DATALOAD = 1								
DOUBLE = 1								
DATAIN%[7:0] = 'h 20	20 28	30	38	30	32			
DATAOUT3[7:0] = 'h 00	00 10	20 24	28 20	30 38 30	32			

DELAY OF DATA BUFFER= 3.7 us

Undo	(u)	Copy	Paste	Belete	ZoomIn	X Zoo	mOutX	ZmOutXFull	Empland
Ti: im End = 1 12-011 -	meA = 0(0) 91,800,000 3 740 153	os Cur Cur Os Time	:sor1 = 3,250 Cursor2 = 7,0 A = 0(0) ps	,847 ps 00,000 ps	0.000.000			100,000,00	10
roup: A	0, 149, 100 .			<u>ים</u> זחחחחחחח	1000,000	10000		<u>100,000,0</u>	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
DATAIN%[7:0] = 'h 2				38		30	32	
	A = 'h x DOUBLE =	x 2	20 21	8	30	38]	J	32
	F = 'h O		10	20 24	28	20 30	38 3	J	32

Design file: /v1/students/dk1ect/AMI4407/stdcell Library Name: linterpol

- 1. Cell: C_UNIT (for Control Unit)
- 2. Cell: mux81 (for Full Custom Multiplexor)
 3. Cell: processor (for Processor)
 4. Cell: dbuffer (for Data Buffer)

- HDL File: /v1/students/dk1ect/AMI4407/stdcell/vhdl