COMPUTER ARCHITECTURE & DESIGN
LAB 12

Objectives

The objective of this lab is to run a program on your machine.

Tasks to be completed

A simple simulated memory is provided to you with a program loaded to run on your machine. You should connect this memory to your machine and run the program. In the unlikely event that your machine does not run perfectly at first, you should find and correct all problems so that the program can run.

Notes

You will find two new files in the Labs folder under resources on the class web site. One file is "SRAM.txt." This is a text file containing the VHDL source to implement the memory. You should create a new VHDL component to realize this memory. Note that the memory does not have a "wait" output because it will always respond within one clock cycle.

The second file is "memoryinit.txt." This file contains the bit vectors used to initialize the low addresses in the memory. You should place this text file in your VHDL working directory. The memory reads this file each time that it is initialized.

A copy of the program is included with this assignment so that you can see the instructions that your machine will need to execute. Each time that your machine writes to the memory, an entry is made in a text file in your working directory. This text file is named "tracefile.txt" and each line has the time, the address, and the data written. When you have your machine operating correctly, you should provide this trace file as part of your report.

Evaluation

All of the labs thus far have been in preparation for this assignment. If you are unable to get the entire machine to work enough to execute the program, it will be very important for you to document what does work, what you have done to try to fix the problems, what you think might be the cause of the problems, and how you would proceed if you had more time.

Your written report for this lab need not be very long, but it should be complete. It should document your entire design as clearly and concisely as you can. It should indicate whether the program finally ran, and if so a copy of the trace file should be included. If you are unable to get the program to run, this is where you should provide the explanation mentioned previously.

Your written report should be in PDF format submitted through the class web site.

Address	Content	Label	Instruction	Comment
0000	0008			Restart Address
0001	0008			Interrupt Address
0002	0008			
0003	0008			
0004	0008			
0005	0008			
0006	0008			
0007	0008			
0008	8000	DIV	LIL R0,0	Initialize R0 to zero
0009	8201		LIL R1,1	Initialize R1 to one
000A	8400		LIL R2,0	
000B	8580		LIH R2,80h	Initialize R2 = 8000h
000C	86FE		LIL R3,FEh	Initialize R3 = FFFEh
000D	8830		LIL R4,30h	
000E	8900		LIH R4,0	Initialize R4 = 0030h (Base of data)
000F	9C18		LIL R14,18h	Initialize R14 = 0018h
0010	8A10		LIL R5,10h	Initialize R5 = 0010h
0011	2C80		LD R6,R4,0	Load R6 from memory [R4] (DIVISOR)
0012	2E81		LD R7,R4,1	Load R7 from memory [R4+1] (ZHIGH)
0013	3082		LD R8,R4,2	Load R8 from memory [R4+2] (ZLOW)
0014	AC00		ADD R6,R0	Set condition codes based on R6
0015	E212		BZ 12h	Branch on zero to ENDLOOP
0016	72C0		MOV R9,R6	Copy R9 = R6
0017	B2E2		SUB R9,R7	Is R9 < R7?
0018	E40F		BN 0Fh	Branch on negative to ENDLOOP
0019	B10C	DLOOP	RL R8,R8	Rotate R8 left
001A	B064		AND R8,R3	Mask R8 LSB to zero
001B	AEEC		RL R7,R7	Rotate R7 left
001C	F004		BC 04h	Branch on carry to SKIP
001D	72E0		MOV R9,R7	Copy R9 = R7
001E	B2C2		SUB R9,R6	Is R9 < R6?
001F	E403		BN 03h	Branch on negative to NOSUB
0020	AEC2	SKIP	SUB R7,R6	R7 = R7 - R6
0021	B020		ADD R8,R1	Add 1 to R8
0022	AA22	NOSUB	SUB R5,R1	Decrement R5
0023	E202		BZ 02h	Branch on zero to END
0024	DC01		JMP R14,1	Jump to DLOOP
0025	4E83	END	ST R4,3,R7	Store R7 in memory [R4+3] (REMAINDER)
0026	5084		ST R4,4,R8	Store R8 in memory [R4+4] (QUOTIENT)
0027	0000	ENDLOOP	NOP	
0028	E0FF		BR FFh	Branch to ENDLOOP
0030	0010	DIVISOR		
0031	0000	ZHIGH		
0032	0100	ZLOW		
0033		REMAINDER		
0034		QUOTIENT		