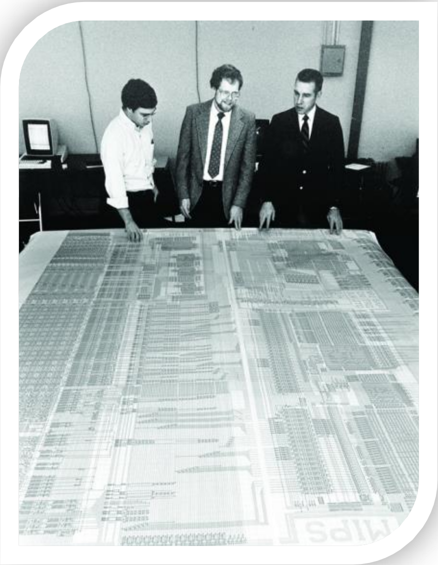


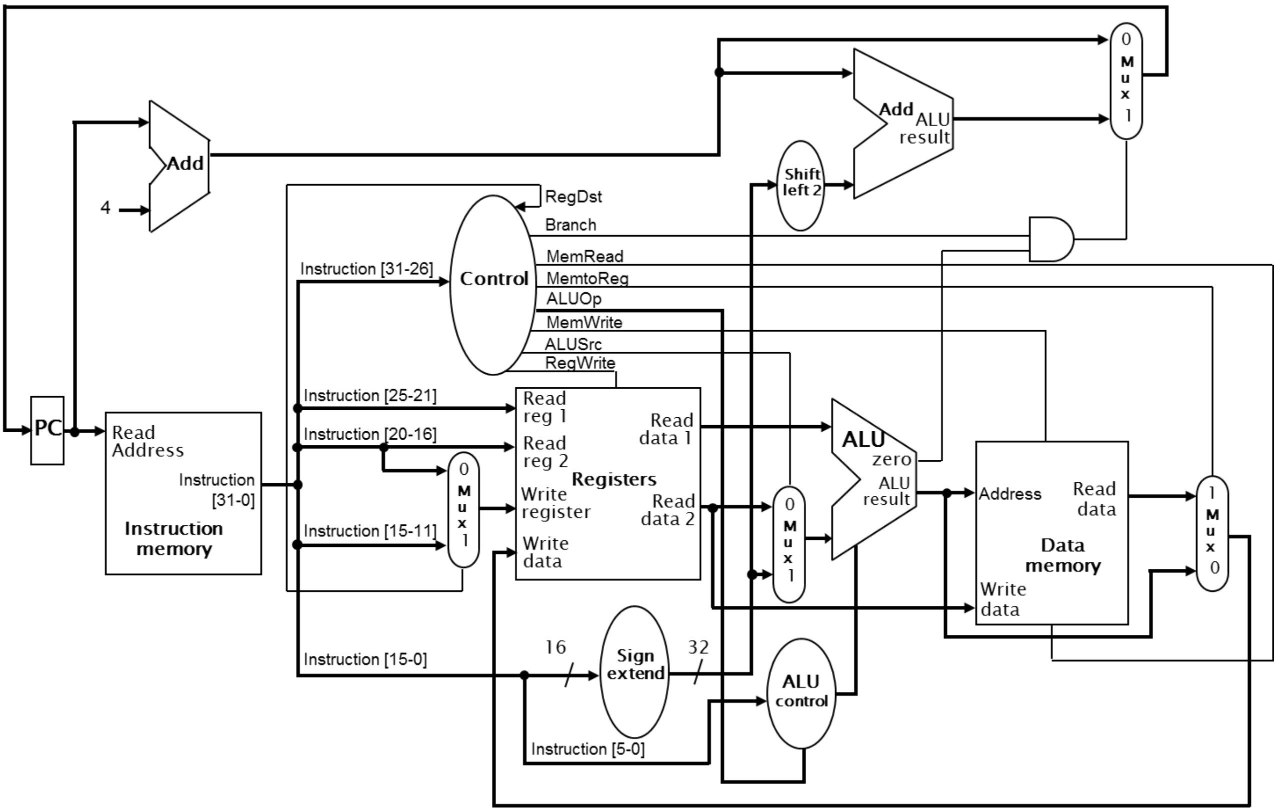
CSE 331 Computer Organizations

Final Project – Fall 2016

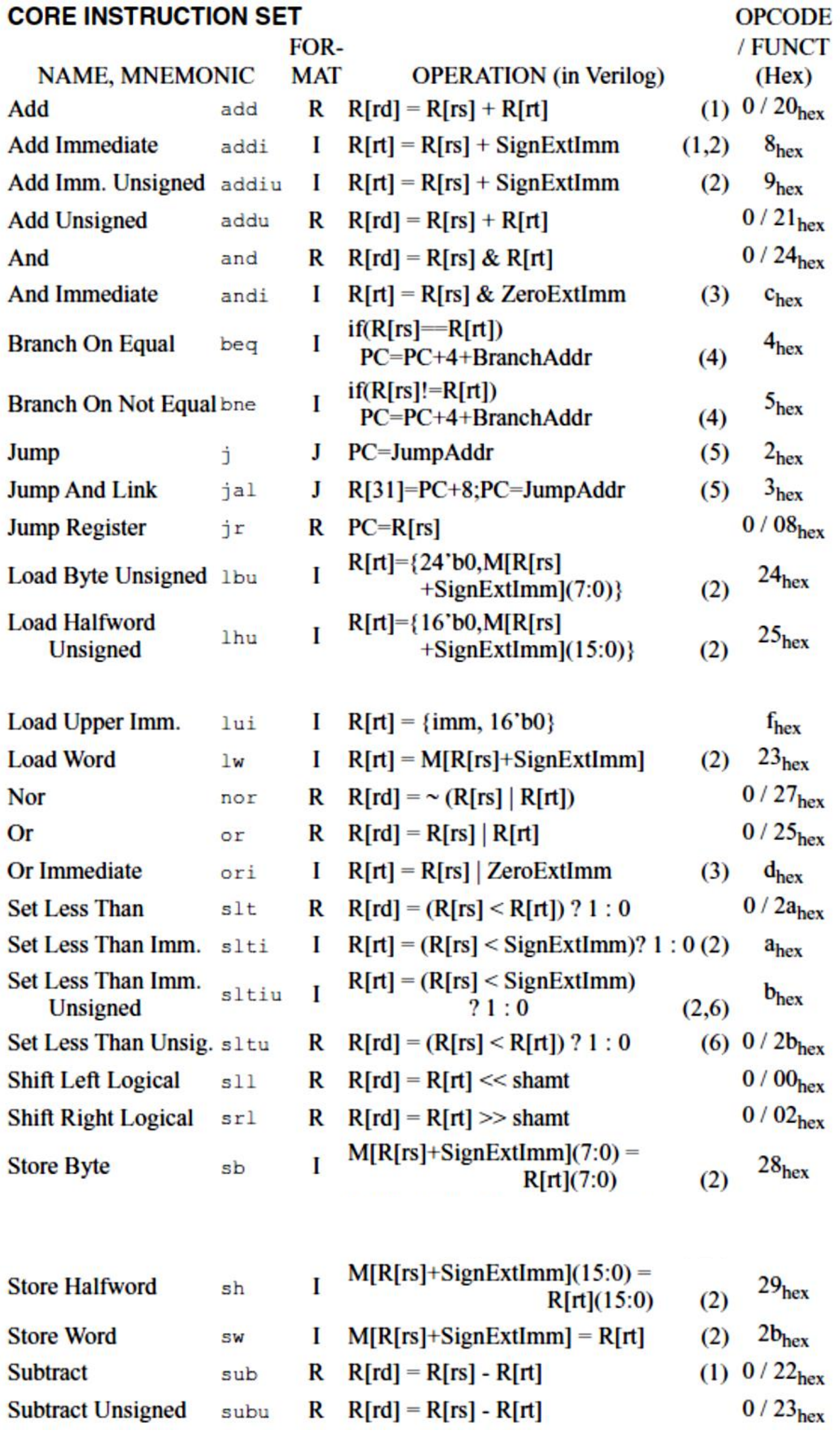
MIPS Processor with Structural Verilog

Due date: December 30, Friday 23:55

In this project you will design an improved version of the single cycle processor shown below using structural Verilog on Altera Quartus II.



Your design will support all core instructions of MIPS shown below:



CORE INSTRUCTION SET

NAME, MNEMONIC		FOR-MAT	OPERATION (in Verilog)		OPCODE / FUNCT (Hex)
Add	add	R	R[rd] = R[rs] + R[rt]	(1)	0 / 20_{hex}
Add Immediate	addi	I	R[rt] = R[rs] + SignExtImm	(1,2)	8_{hex}
Add Imm. Unsigned	addiu	I	R[rt] = R[rs] + SignExtImm	(2)	9_{hex}
Add Unsigned	addu	R	R[rd] = R[rs] + R[rt]		0 / 21_{hex}
And	and	R	R[rd] = R[rs] & R[rt]		0 / 24_{hex}
And Immediate	andi	I	R[rt] = R[rs] & ZeroExtImm	(3)	c_{hex}
Branch On Equal	beq	I	if(R[rs]==R[rt]) PC=PC+4+BranchAddr	(4)	4_{hex}
Branch On Not Equal	bne	I	if(R[rs]!=R[rt]) PC=PC+4+BranchAddr	(4)	5_{hex}
Jump	j	J	PC=JumpAddr	(5)	2_{hex}
Jump And Link	jal	J	R[31]=PC+8;PC=JumpAddr	(5)	3_{hex}
Jump Register	jr	R	PC=R[rs]		0 / 08_{hex}
Load Byte Unsigned	lbu	I	R[rt]={24'b0,M[R[rs] +SignExtImm](7:0)}	(2)	24_{hex}
Load Halfword Unsigned	lhu	I	R[rt]={16'b0,M[R[rs] +SignExtImm](15:0)}	(2)	25_{hex}
Load Upper Imm.	lui	I	R[rt] = {imm, 16'b0}		f_{hex}
Load Word	lw	I	R[rt] = M[R[rs]+SignExtImm]	(2)	23_{hex}
Nor	nor	R	R[rd] = ~ (R[rs] | R[rt])		0 / 27_{hex}
Or	or	R	R[rd] = R[rs] | R[rt]		0 / 25_{hex}
Or Immediate	ori	I	R[rt] = R[rs] | ZeroExtImm	(3)	d_{hex}
Set Less Than	slt	R	R[rd] = (R[rs] < R[rt]) ? 1 : 0		0 / $2a_{hex}$
Set Less Than Imm.	slti	I	R[rt] = (R[rs] < SignExtImm)? 1 : 0	(2)	a_{hex}
Set Less Than Imm. Unsigned	sltiu	I	R[rt] = (R[rs] < SignExtImm) ? 1 : 0	(2,6)	b_{hex}
Set Less Than Unsig.	sltu	R	R[rd] = (R[rs] < R[rt]) ? 1 : 0	(6)	0 / $2b_{hex}$
Shift Left Logical	sll	R	R[rd] = R[rt] << shamt		0 / 00_{hex}
Shift Right Logical	srl	R	R[rd] = R[rt] >> shamt		0 / 02_{hex}
Store Byte	sb	I	M[R[rs]+SignExtImm](7:0) = R[rt](7:0)	(2)	28_{hex}
Store Halfword	sh	I	M[R[rs]+SignExtImm](15:0) = R[rt](15:0)	(2)	29_{hex}
Store Word	sw	I	M[R[rs]+SignExtImm] = R[rt]	(2)	$2b_{hex}$
Subtract	sub	R	R[rd] = R[rs] - R[rt]	(1)	0 / 22_{hex}
Subtract Unsigned	subu	R	R[rd] = R[rs] - R[rt]		0 / 23_{hex}

Rules:

1. Behavioral Verilog is NOT PERMITTED. Thus, first draw your schematic on a paper. Only Structural Verilog is allowed. (All your Boolean expressions will be implemented by **`assign`**, etc. But you cannot use **`always`**, **`if`**, **`case`**, etc.)

2. Designs that are not even simulating can at most get 30pts. You get partial grades if at least a reasonable subset of the instructions shown above can be executed.

3. All project details will be announced at next PS (December 5). So attend the PS for your own good! Otherwise you get no credits.

4. The input-output names must be exactly the same as explained by the instructor during the PS.

5. You should also write Verilog testbench and simulate your design using ModelSim. Use writememh and readmemh for instruction memory initialization and content viewing of the memory.

6. Each day of late submission will result in 15pts reduction.

7. This project is REALLY IMPORTANT FOR YOU. So please ATTEND whatever it takes.

8. Feel free to ask if you have any difficulty.

9. (BONUS) If you come with a solution to perform a successful demo on Terasic DE0 board, you will get 25pts Bonus. Ask details to your TA.

10. Start today. Tomorrow may be too late.

11. Honor code: It is not a group project. Any cheating means at least -100 for both sides. Do not share your codes and design to any one in any circumstance. Be honest and uncorrupt not to win but because it is right!

PROF. SMITH, MY PROJECT DOESN'T WORK.
I SEE. HOW MANY TIMES HAVE YOU FAILED?
TWICE.
HMMMM.
IS THAT ENOUGH?
FAIL EIGHT MORE TIMES AND CALL ME IN THE MORNING.
JORGE CHAM ©2015
WWW.PHDCOMICS.COM