

P1. (10 points): Explain the meaning of this register machine program. If the initial values of the registers are $R1=5$, $R2=3$, $R3=6$, and $R4=9$, what will be their final values?

STEP	INSTRUCTION	REGISTER	GO TO STEP	[BRANCH TO STEP]
1.	Deb	2	2	4
2.	Inc	1	3	
3.	Inc	3	1	
4.	End			

P2. (15 points) Type the program from page 115 in the assigned reading on register machines into RodRego: <https://rodrego.it.tufts.edu/>
 Then, run the program for two different input conditions (different values stored in the registers). Provide several screenshots of the execution of your program.

P3. (15 points) Consider a register machine with four registers $R0$, $R1$, $R2$, and $R3$. Let the initial values be: $R0 = x$, $R1 = y$, $R2 = z$, $R3=0$. The values x , y , and z are two random integers (positive or zero). Write the instructions (in the table format shown in P1) for a register machine program that computes $3x + 2y$ and store the result in $R3$. Write a short comment for each line/block of your program.

P4. (15 points) Decoding a CPU instruction.

- Draw the circuit diagram for the OpCode decoder of the i281 CPU using decoders of various sizes. Label all inputs, outputs, and pins.
- Explain how the output SHIFTL is computed.
- Give concrete bit values for a 16-bit instruction that multiplies register D by 2.

P5. (15 points) Computing the new program counter value.

- Draw the circuit diagram that updates the program counter using two 8-bit adders, a bus multiplexer, and a register. Label all inputs, outputs, and pins.
- Explain how the PC is computed for a JUMP instruction.
- Explain how the PC is computed for a BRE instruction.

P6. (10 points) The i281 assembler is written in Java. It takes an assembly program, which is stored in a plain text file with asm extension, and maps it to machine code. Download the sample assembly programs and the assembler from this link:
https://www.ece.iastate.edu/~alexs/classes/i281_CPU/i281_CPU_Software.zip

Read the README.txt for instructions. Pick one of the sample programs and compile it (pick a file that is not in the instructions). To get the 10 points, attach one or more screenshots of the text that the assembler prints on the screen.

P7. (10 points) Write an assembly program (in i281 assembly language) that loads the variables X=3, Y=2, and Z=5 from data memory into registers A, B, and C, respectively. Then, computes $3A + 2B - C$ and stores the result in register D. Finally, it overwrites the value of Z with the contents of register D.

P8. (10 points) Convert your assembly program from P7 into i281 machine code. You can do this by hand or with the java assembler. To get the 10 points, however, you must use the i281 simulator, which works in a web browser (for best results use Firefox):
https://www.ece.iastate.edu/~alexs/classes/i281_simulator/index.html

Hint: Click on the “Load” button to try some of the stored examples. You may find the ones in the Arithmetic section particularly useful. Click the green button labeled “Go to CPU” and then “Step” multiple times to trace the execution of the program. You can also try the “Load New File” button to upload your own assembly file (in plain text format) into the simulator.

Provide a screenshot of your program loaded into the simulator with the syntax highlighting turned on. Your image should look like this one:

Assembly Code:				Machine Code:	
.data				View Data Memory	
0	x	BYTE	2		
1	z	BYTE	?		
.code				Instruction Memory:	
0	LOAD	A,	[x]	1000_00_00_00000000	
1	MOVE	C,	A	0010_10_00_00000000	
2	ADDI	C,	3	0101_10_00_00000011	
3	STORE	[z],	C	1010_10_00_00000001	