EE 224 - Digital Systems
Course Project
Design a Simple i281 Processor (Toy-CPU)

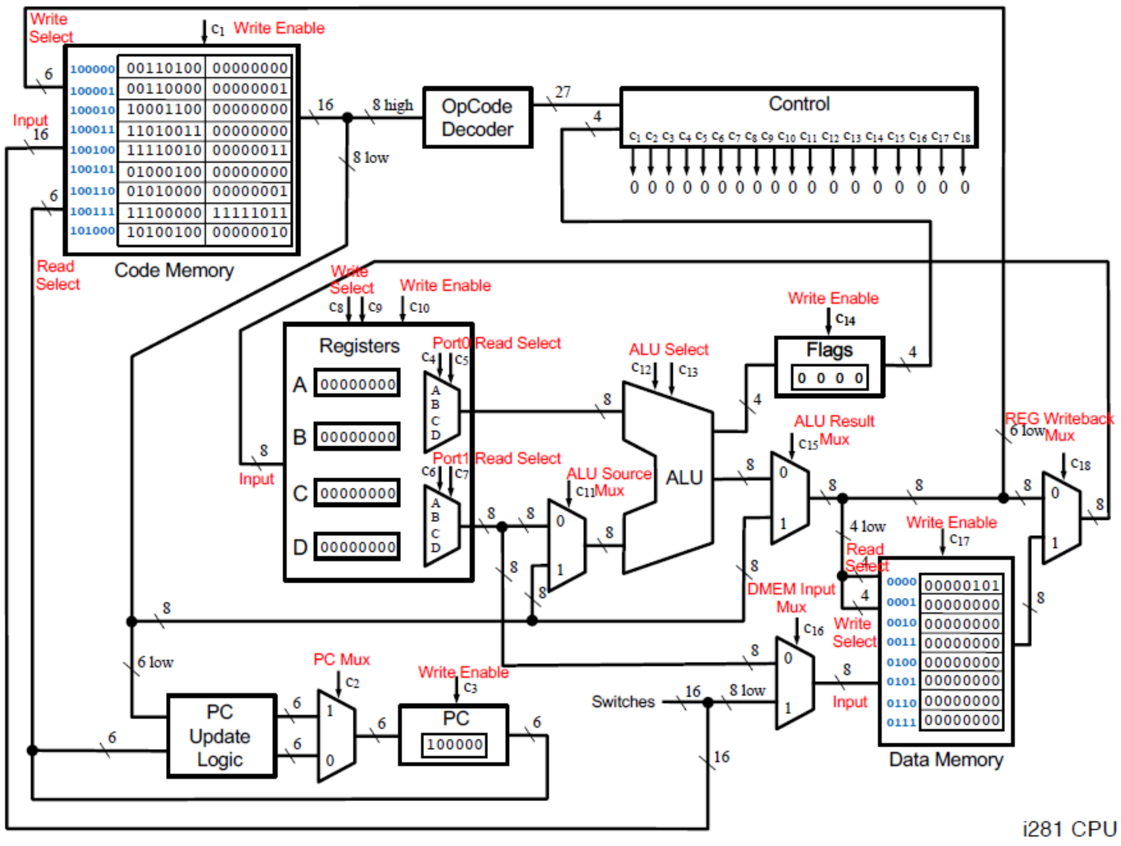
Problem Statement

Objective

Design and implement a simplified CPU capable of executing a **bubble sort** algorithm. Your processor should be able to read an unsorted array from memory, sort it, and write the sorted result back to memory. **(3 students per team)**

CPU Specifications

1. Architecture Overview



2. Opcodes

The i281 Assembly Instructions

NOOP	NO OPeration
INPUTC	INPUT into Code memory
INPUTCF	INPUT into Code memory with oFfset
INPUTD	INPUT into Data memory
INPUTDF	INPUT into Data memory with oFfset
MOVE	MOVE the contents of one register into another
LOADI	LOAD Immediate value
LOADP	LOAD Pointer address
ADD	ADD two registers
ADDI	ADD an Immediate value to a register
SUB	SUBtract two registers
SUBI	SUBtract an Immediate value from a register
LOAD	LOAD from a data memory address into a register
LOADF	LOAD with an oFfset specified by another register
STORE	STORE a register into a data memory address
STOREF	STORE with an oFfset specified by another register
SHIFTL	SHIFT Left all bits in a register
SHIFTR	SHIFT Right all bits in a register
CMP	CoMPare the values in two registers
JUMP	JUMP unconditionally to a specified address
BRE	BRanch if Equal
BRZ	BRanch if Zero
BRNE	BRanch if Not Equal
BRNZ	BRanch if Not Zero
BRG	BRanch if Greater
BRGE	BRanch if Greater than or Equal

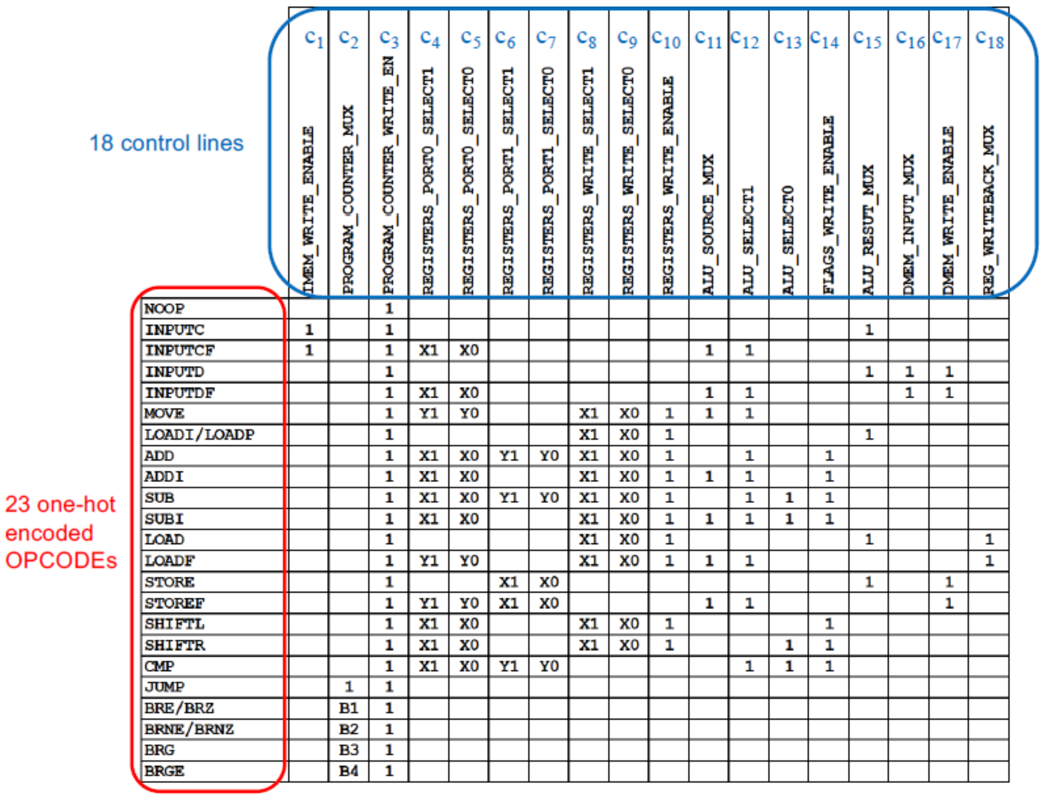
The OPCODEs

NOOP	0	0	0	0	d	d	d	d	d	d	d	d	d	d	d	d
INPUTC	0	0	0	1	d	d	0	0	C	A	D	D	R	E	S	S
INPUTCF	0	0	0	1	R	X	0	1	C	A	D	D	R	E	S	S
INPUTD	0	0	0	1	d	d	1	0	D	A	D	D	R	E	S	S
INPUTDF	0	0	0	1	R	X	1	1	D	A	D	D	R	E	S	S
MOVE	0	0	1	0	R	X	R	Y	0	0	0	0	0	0	0	0
LOADI/LOADP	0	0	1	1	R	X	d	d	I	M	M	E	D	V	A	L

ADD	0	1	0	0	R	X	R	Y	d	d	d	d	d	d	d	d
ADDI	0	1	0	1	R	X	d	d	I	M	M	E	D	V	A	L
SUB	0	1	1	0	R	X	R	Y	d	d	d	d	d	d	d	d
SUBI	0	1	1	1	R	X	d	d	I	M	M	E	D	V	A	L
LOAD	1	0	0	0	R	X	d	d	D	A	D	D	R	E	S	S
LOADF	1	0	0	1	R	X	R	Y	D	A	D	D	R	E	S	S
STORE	1	0	1	0	R	X	d	d	D	A	D	D	R	E	S	S
STOREF	1	0	1	1	R	X	R	Y	D	A	D	D	R	E	S	S
SHIFTL	1	1	0	0	R	X	d	0	d	d	d	d	d	d	d	d
SHIFTR	1	1	0	0	R	X	d	1	d	d	d	d	d	d	d	d
CMP	1	1	0	1	R	X	R	Y	d	d	d	d	d	d	d	d
JUMP	1	1	1	0	d	d	d	d	P	C	O	F	F	S	E	T
BRE/BRZ	1	1	1	1	d	d	0	0	P	C	O	F	F	S	E	T
BRNE/BRNZ	1	1	1	1	d	d	0	1	P	C	O	F	F	S	E	T
BRG	1	1	1	1	d	d	1	0	P	C	O	F	F	S	E	T
BRGE	1	1	1	1	d	d	1	1	P	C	O	F	F	S	E	T

Note: d represents "don't care" bits

3. Instruction decode logic


18 control lines
23 one-hot encoded OPCODEs

	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_{12}	c_{13}	c_{14}	c_{15}	c_{16}	c_{17}	c_{18}
	IMEM_WRITE_ENABLE	PROGRAM_COUNTER_MUX	PROGRAM_COUNTER_WRITE_EN	REGISTERS_PORT0_SELECT1	REGISTERS_PORT0_SELECT0	REGISTERS_PORT1_SELECT1	REGISTERS_PORT1_SELECT0	REGISTERS_WRITE_SELECT1	REGISTERS_WRITE_SELECT0	REGISTERS_WRITE_ENABLE	ALU_SOURCE_MUX	ALU_SELECT1	ALU_SELECT0	FLAGS_WRITE_ENABLE	ALU_RESUT_MUX	DMEM_INPUT_MUX	DMEM_WRITE_ENABLE	REG_WRITEBACK_MUX
NOOP			1															
INPUTC	1		1												1			
INPUTCF	1		1	X1	X0						1	1						
INPUTD			1												1	1	1	
INPUTDF			1	X1	X0						1	1				1	1	
MOVE			1	Y1	Y0			X1	X0	1	1	1						
LOADI/LOADP			1					X1	X0	1					1			
ADD			1	X1	X0	Y1	Y0	X1	X0	1		1		1				
ADDI			1	X1	X0			X1	X0	1	1	1		1				
SUB			1	X1	X0	Y1	Y0	X1	X0	1		1	1	1				
SUBI			1	X1	X0			X1	X0	1	1	1	1	1				
LOAD			1					X1	X0	1					1			1
LOADF			1	Y1	Y0			X1	X0	1	1	1						1
STORE			1			X1	X0								1		1	
STOREF			1	Y1	Y0	X1	X0				1	1					1	
SHIFTL			1	X1	X0			X1	X0	1				1				
SHIFTR			1	X1	X0			X1	X0	1			1	1				
CMP			1	X1	X0	Y1	Y0					1	1	1				
JUMP		1	1															
BRE/BRZ		B1	1															
BRNE/BRNZ		B2	1															
BRG		B3	1															
BRGE		B4	1															

Submission Guidelines

1. Submit all design files (Verilog or VHDL)in a single compressed zip folder.
2. Include a REPORT with:
   - Team member names
   - File structure description
   - Simulation waveforms showing register file contents and data memory contents, and flags for necessary iterations.

Resources and References

- Refer to the provided lecture slides on CPU architecture and assembly language
- Study the i281 CPU architecture as a reference design
- Review the ALU and Program Counter implementation details
- Consult the instruction encoding tables in the course materials