RISC-V Single Cycle Processor - Instruction Flow

Clock Cycle by Cycle Analysis

Cycle 0: Reset

- **PC = 0**
- All registers initialized to test values
- inst_out = 0x00000000 (NOP)

Cycle 1: ADD x13, x16, x25

PC = 4 | inst_out = `0000000_11001_10000_000_01101_0110011`

Stage	Signal	Value	Description
Decode	opcode	0110011	R-type
	rs1	10000 (x16)	Source register 1
	rs2	11001 (x25)	Source register 2
	rd	01101 (x13)	Destination register
	funct3	000	ADD operation
	funct7	0000000	ADD (not SUB)
Control	ALU_src	0	Use rs2 data
	mem_to_reg	0	Use ALU result
	reg_wr	1	Write to register
	ALU_op	10	R-type operation
Register Read	read_data1	40	RegFile[16]
	read_data2	65	RegFile[25]
ALU	Ctrl_out	0010	ADD operation
	ALU_result	105	40 + 65
	zero	0	Not zero
Write Back	wr_data	105	Write to x13

Result: RegFile[13] = 105

Cycle 2: SUB x5, x8, x3

PC = 8 | inst_out = `0100000_00011_01000_000_00101_0110011`

Stage	Signal	Value	Description
Decode	rs1	01000 (x8)	
	rs2	00011 (x3)	
	rd	00101 (x5)	
	funct7	0100000	SUB indicator
Register Read	read_data1	2	RegFile[8]
	read_data2	24	RegFile[3]
ALU	Ctrl_out	0110	SUB operation
	ALU_result	-22 (0xFFFFFFEA)	2 - 24
Write Back	wr_data	-22	Write to x5

Result: RegFile[5] = -22

Cycle 3: AND x1, x2, x3

PC = 12 | inst_out = `0000000_00011_00010_111_00001_0110011`

Stage	Signal	Value	Description
Decode	rs1	00010 (x2)	
	rs2	00011 (x3)	
	rd	00001 (x1)	
	funct3	111	AND operation
Register Read	read_data1	2	RegFile[2]
	read_data2	24	RegFile[3]
ALU	Ctrl_out	0000	AND operation
	ALU_result	0	2 & 24 = 0
Write Back	wr_data	0	Write to x1

Result: RegFile[1] = 0

Cycle 4: OR x4, x3, x5

PC = 16 | inst_out = `0000000_00101_00011_110_00100_0110011`

Stage	Signal	Value	Description
Decode	rs1	00011 (x3)	
	rs2	00101 (x5)	
	rd	00100 (x4)	
	funct3	110	OR operation
Register Read	read_data1	24	RegFile[3]
	read_data2	-22	RegFile[5] (updated!)
ALU	Ctrl_out	0001	OR operation
	ALU_result	-2 (0xFFFFFFFE)	24
Write Back	wr_data	-2	Write to x4

Result: RegFile[4] = -2

Cycle 5: ADDI x22, x21, 3

PC = 20 | inst_out = `000000000011_10101_000_10110_0010011`

Stage	**Signal**	**Value**	**Description**
Decode	opcode	0010011	I-type
	rs1	10101 (x21)	
	rd	10110 (x22)	
	imm[11:0]	000000000011	3
Control	ALU_src	1	Use immediate
	ALU_op	11	I-type operation
Register Read	read_data1	80	RegFile[21]
Immediate Gen	immediate_ext	3	Sign-extended
ALU	A	80	rs1 value
	B	3	Immediate
	ALU_result	83	80 + 3
Write Back	wr_data	83	Write to x22

Result: RegFile[22] = 83

Cycle 6: ORI x9, x8, 1

PC = 24 | inst_out = `000000000001_01000_110_01001_0010011`

Stage	Signal	Value	Description
Decode	rs1	01000 (x8)	
	rd	01001 (x9)	
	imm	1	
	funct3	110	ORI operation
Register Read	read_data1	2	RegFile[8]
ALU	Ctrl_out	0001	OR operation
	ALU_result	3	2
Write Back	wr_data	3	Write to x9

Result: RegFile[9] = 3

Cycle 7: LW x8, 15(x5)

PC = 28 | inst_out = `000000001111_00101_010_01000_0000011`

Stage	Signal	Value	Description
Decode	opcode	0000011	Load
	rs1	00101 (x5)	Base address
	rd	01000 (x8)	
	imm	15	Offset
Control	ALU_src	1	Use immediate
	mem_rd	1	Read memory
	mem_to_reg	1	Use memory data
Register Read	read_data1	-22	RegFile[5]
ALU	ALU_result	-7	-22 + 15
Data Memory	addr	-7	Memory address
	rd_data	0	D_memory[-7]
Write Back	wr_data	0	From memory

Result: RegFile[8] = 0

Cycle 8: LW x9, 3(x3)

PC = 32 | inst_out = 000000000011_00011_010_01001_0000011

Stage	Signal	Value	Description
Register Read	read_data1	24	RegFile[3]
ALU	ALU_result	27	24 + 3
Data Memory	addr	27	Memory address
	rd_data	0	D_memory[27]
Write Back	wr_data	0	From memory

Result: RegFile[9] = 0

Cycle 9: SW x15, 12(x5)

PC = 36 | inst_out = 0000000_01111_00101_010_01100_0100011

Stage	Signal	Value	Description
Decode	opcode	0100011	Store
	rs1	00101 (x5)	Base address
	rs2	01111 (x15)	Data to store
	imm	12	Offset
Control	mem_wr	1	Write memory
	reg_wr	0	No register write
Register Read	read_data1	-22	RegFile[5]
	read_data2	30	RegFile[15]
ALU	ALU_result	-10	-22 + 12
Data Memory	addr	-10	Memory address
	wr_data	30	Store value

Result: D_memory[-10] = 30

Cycle 10: SW x14, 10(x6)

PC = 40 | inst_out = 0000000_01110_00110_010_01010_0100011

Stage	Signal	Value	Description
Register Read	read_data1	44	RegFile[6]
	read_data2	20	RegFile[14]
ALU	ALU_result	54	44 + 10
Data Memory	wr_data	20	Store value

Result: D_memory[54] = 20

Summary Table: Register State After Execution

Register	Initial	After Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5	Cycle 6	Cycle 7	Cycle 8
x1	4	4	4	**0**	0	0	0	0	0
x4	4	4	4	4	**-2**	-2	-2	-2	-2
x5	1	1	**-22**	-22	-22	-22	-22	-22	-22
x8	2	2	2	2	2	2	2	**0**	0
x9	1	1	1	1	1	1	**3**	3	**0**
x13	10	**105**	105	105	105	105	105	105	105
x22	90	90	90	90	90	**83**	83	83	83

Datapath Signal Flow

```
PC → Inst_Memory → Instruction[31:0]
              ↓
       [Decode Instruction]
              ↓
   rs1[19:15]  rs2[24:20]  rd[11:7]
        ↓           ↓          ↓
   Register File ─────────→  Write Data (from mux3)
        ↓           ↓
   read_data1  read_data2
        ↓           ↓
        ↓      ┌───────┴────────┐
        ↓      │   MUX1    │← immediate (if ALU_src=1)
        ↓      └───────┬────────┘
        ↓           ↓
        └──→   ALU  ←──────── ALU_Control
              ↓
         ALU_result ─────────→ Data_Memory (if store)
              ↓             ↓
         ┌───────┴──────────┴──────────┘
         │   MUX3    │← mem_data (if mem_to_reg=1)
         └───────┬──────────┘
              ↓
         Write Data → Register File[rd]
```