

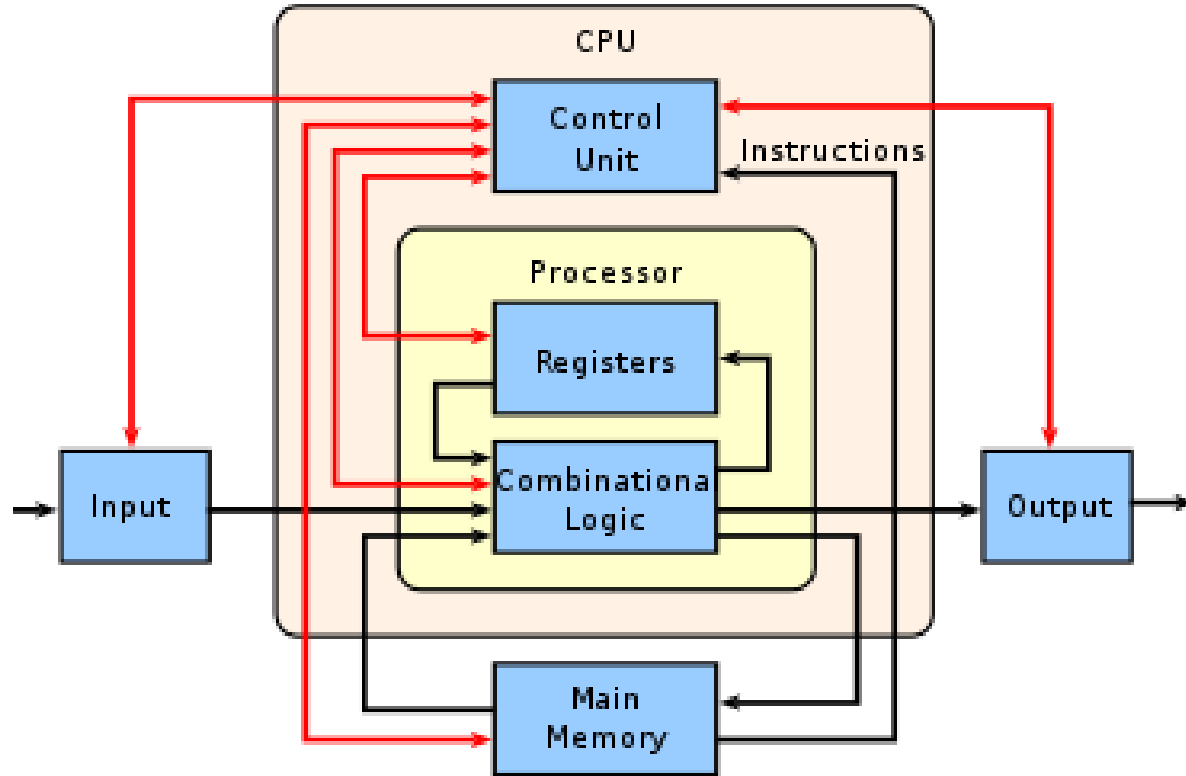
WSIZ, 2023
Computer systems architecture

Lecture 13. Microarchitecture. Typical processor design.

Dmitry Zaitsev

<http://daze.ho.ua>

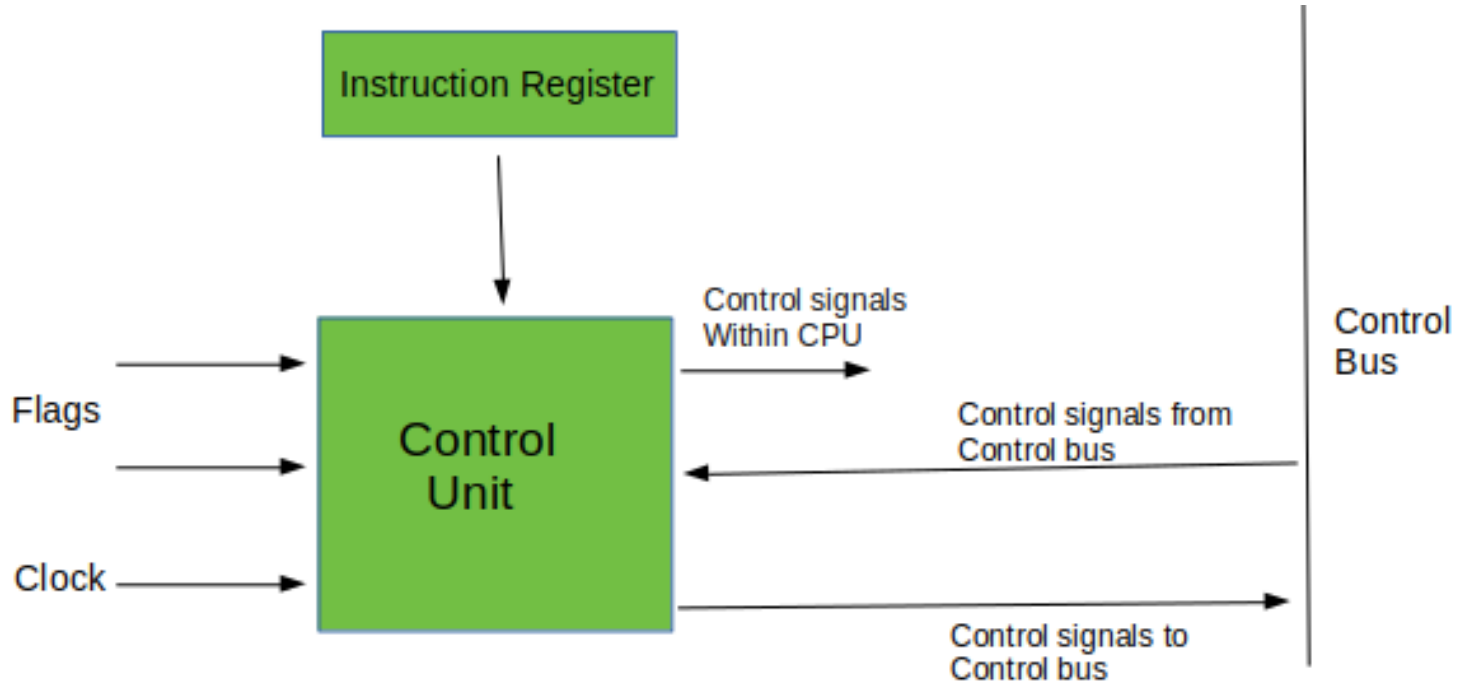
CPU structure



CPU basic components

- **Control unit (CU)** directs the operation of the processor.
- **Arithmetic logic unit (ALU)** performs arithmetic and other operations.
- **Registers** store data.

Control unit connection



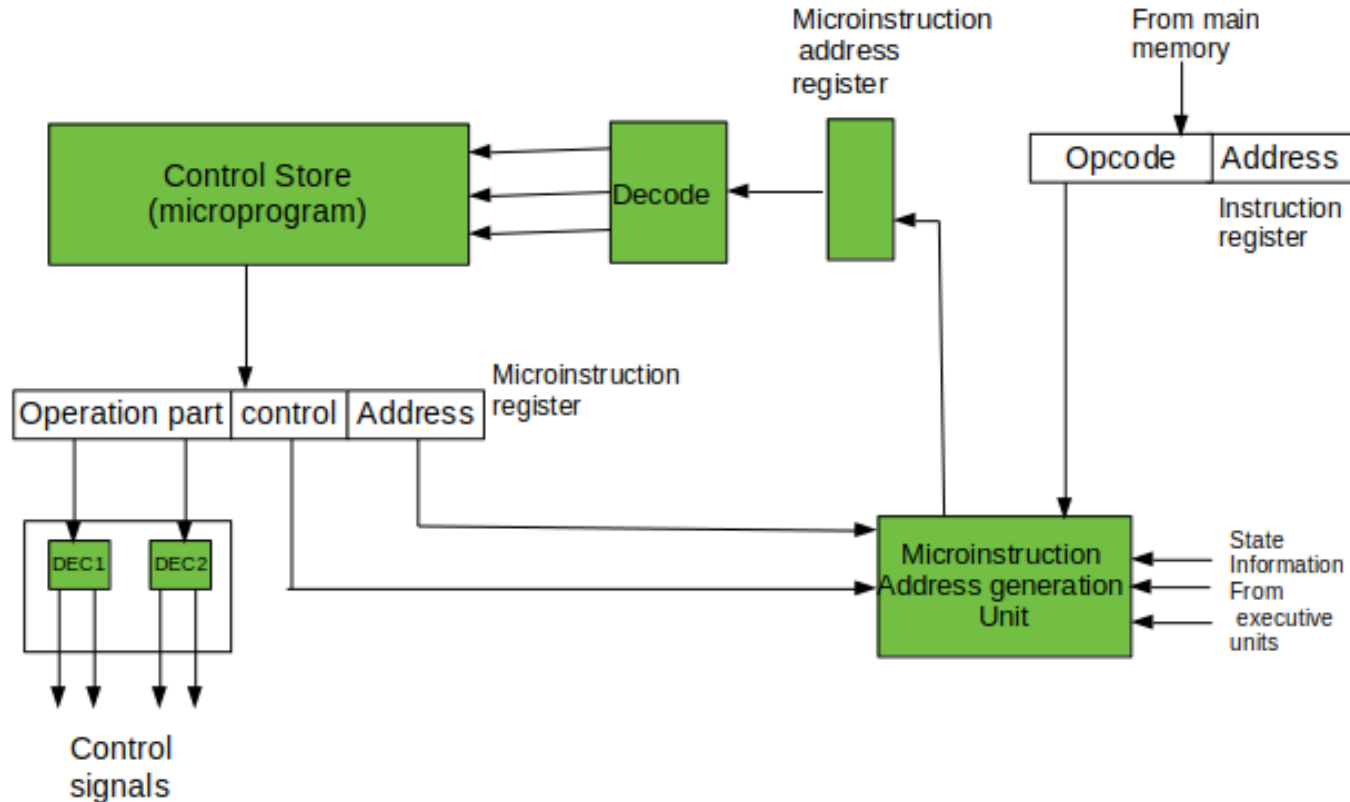
Control unit functions

- Coordinates the sequence of data movements into, out of, and between a processor's many sub-units.
- Interprets instructions.
- Controls data flow inside the processor.
- Receives external instructions or commands to which it converts to sequence of control signals.
- Controls execution units (i.e. ALU, data buffers and registers) contained within a CPU.
- Handles multiple tasks, such as fetching, decoding, execution handling and storing results.

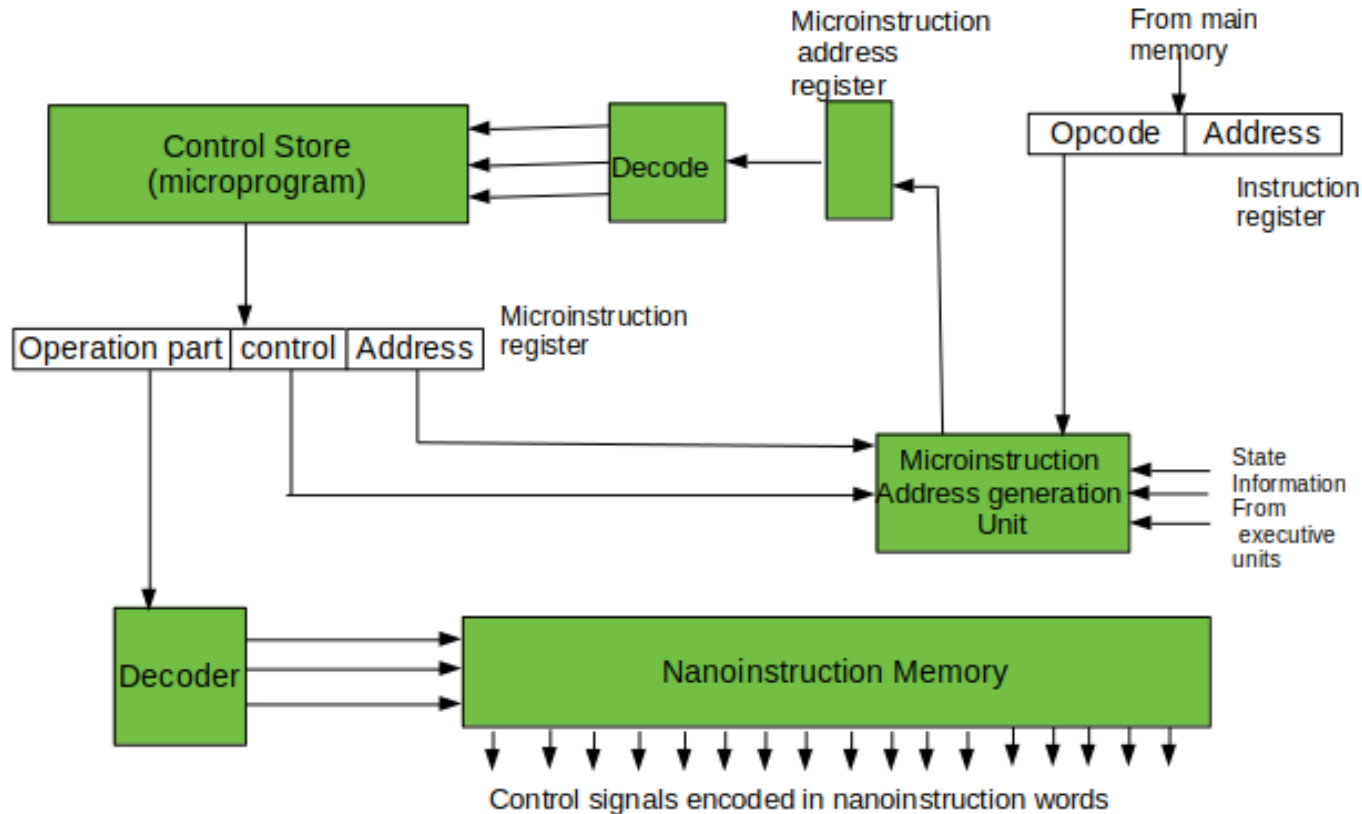
Types of control unit

- **Hardwired Control Unit** – the control signals are generated by specially designed hardware logical circuits
- **Microprogrammable control unit (MCU)** – based on the control store that is used for storing words containing encoded control signals.
 - With a single-level control store
 - With a two-level control store

MCU with a single-level control store



MCU with a two-level control store



Prototyping a processor

- RISC – Reduced Instruction Set Computer
- Separate memory for instructions and data
- Instruction with 3 operands: $o1 \leftarrow o2 * o3$
- Operand addressing mode: register, immediate, based

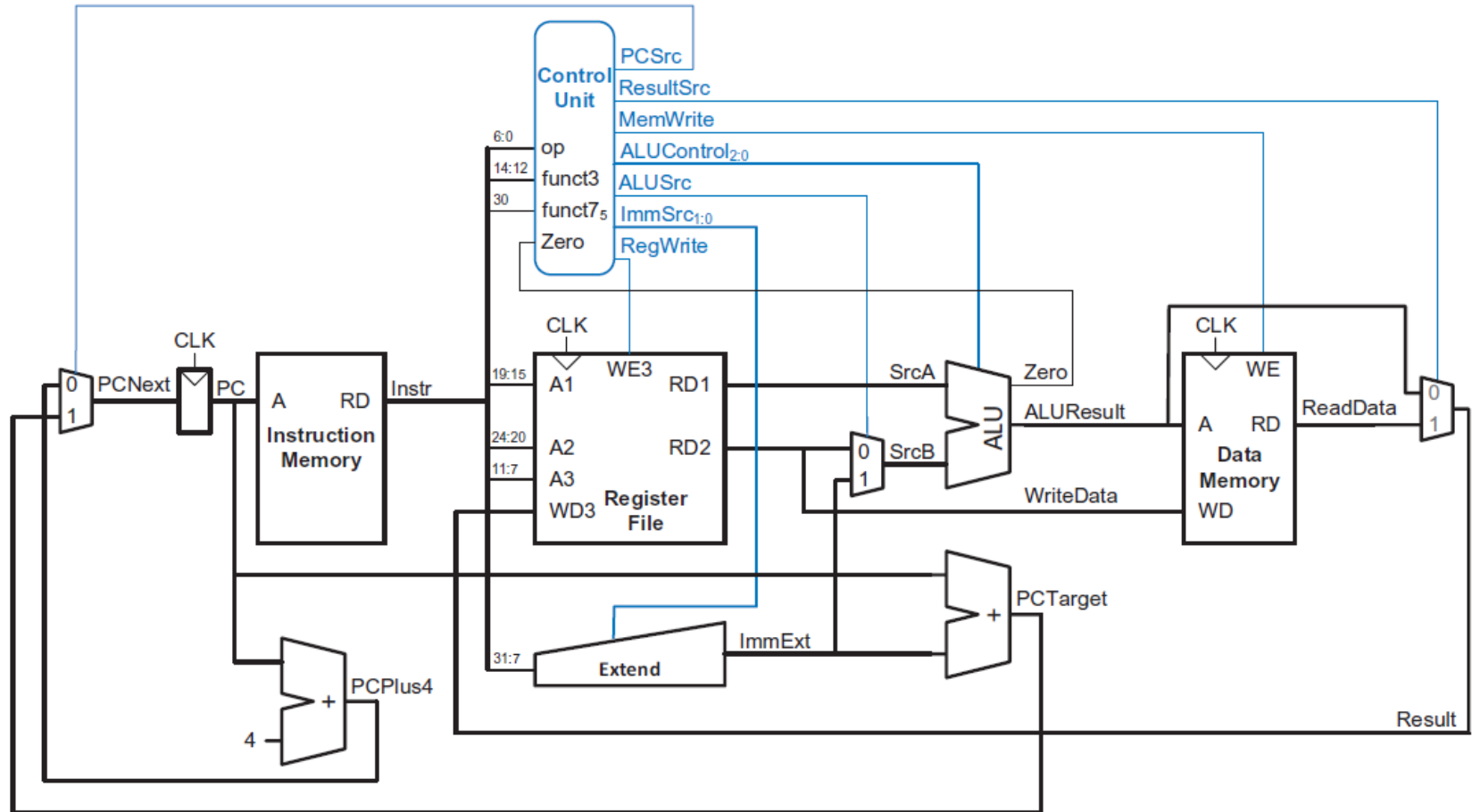
Processor architecture

- Instructions: add, sub, and, or, lw, sw, beq,...
- Buses: address – 32 bits, data – 32 bits
- Registers: x0-x31
- Single time-cycle

Registers of processor

Name	Register Number	Use
zero	x0	Constant value 0
ra	x1	Return address
sp	x2	Stack pointer
gp	x3	Global pointer
tp	x4	Thread pointer
t0-2	x5-7	Temporary registers
s0/fp	x8	Saved register/Frame pointer
s1	x9	Saved register
a0-1	x10-11	Function arguments/Return values
a2-7	x12-17	Function arguments
s2-11	x18-27	Saved registers
t3-6	x28-31	Temporary registers

Single cycle processor



Blocks and signals (1)

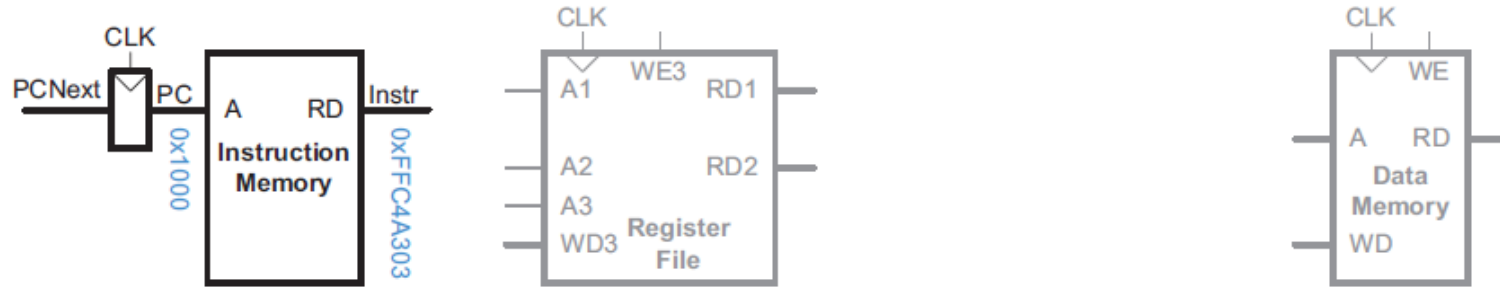
- PC (program counter); *PCNext* the address of the next instruction
- *Instruction memory*; instruction address input, *A*, instruction output, *RD*
- *Register file* holds registers x0–x31; 5-bit address input: *A1*, *A2* to read, *A3* to write when *WE3* is on; *RD1*, *RD2* read data; *WD3* write data

Blocks and signals (2)

- *Extend* extends immediate operand (inside instruction) to 32 bits
- *Data memory*; *A* address; *RD* read data; *WD* write data when *WE* is enabled

Trace instruction execution

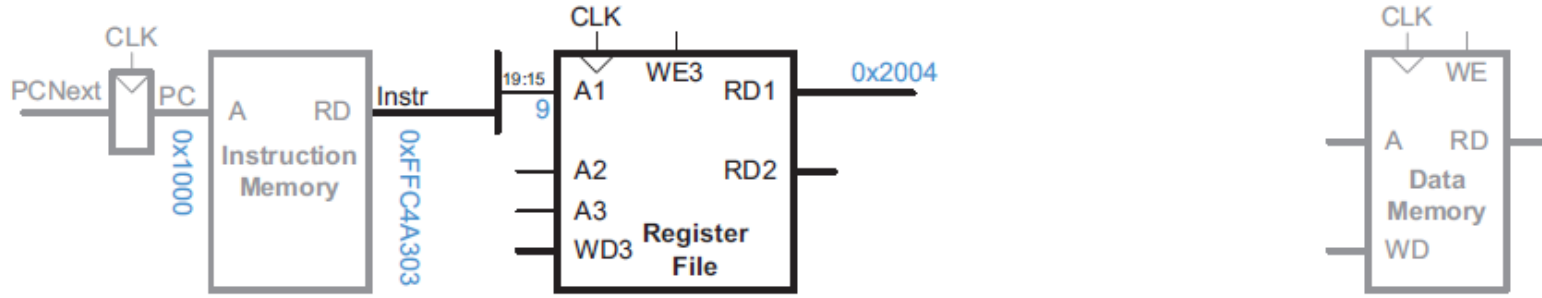
(a) fetch instruction from memory



Address	Instruction	Type	Fields	Machine Language
0x1000	L7: lw x6, -4(x9)	I	<div> $imm_{11:0}$ </div> <div> $rs1$ </div> <div> $f3$ </div> <div> rd </div>	<div> op </div> <div>FFC4A303</div>
			<div>111111111100</div> <div>01001</div> <div>010</div> <div>00110</div>	<div>0000011</div>

Trace instruction execution

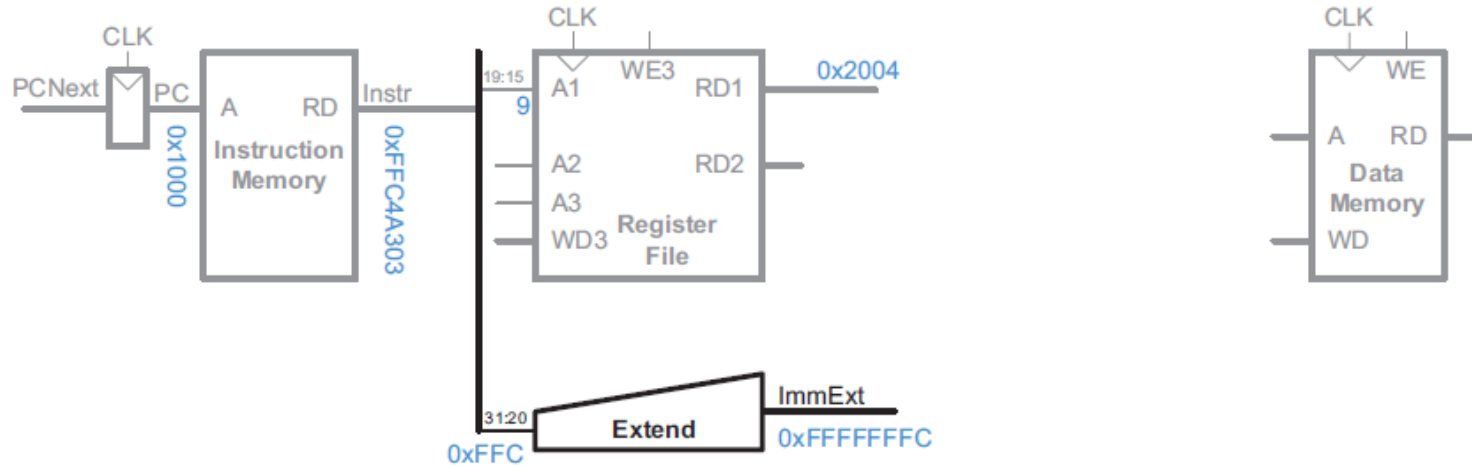
(b) read source operand from register file



Address	Instruction	Type	Fields			Machine Language	
0x1000	L7: lw x6, -4(x9)	I	imm _{11:0}	rs1	f3	rd	op
			111111111100	01001	010	00110	0000011
							FFC4A303

Trace instruction execution

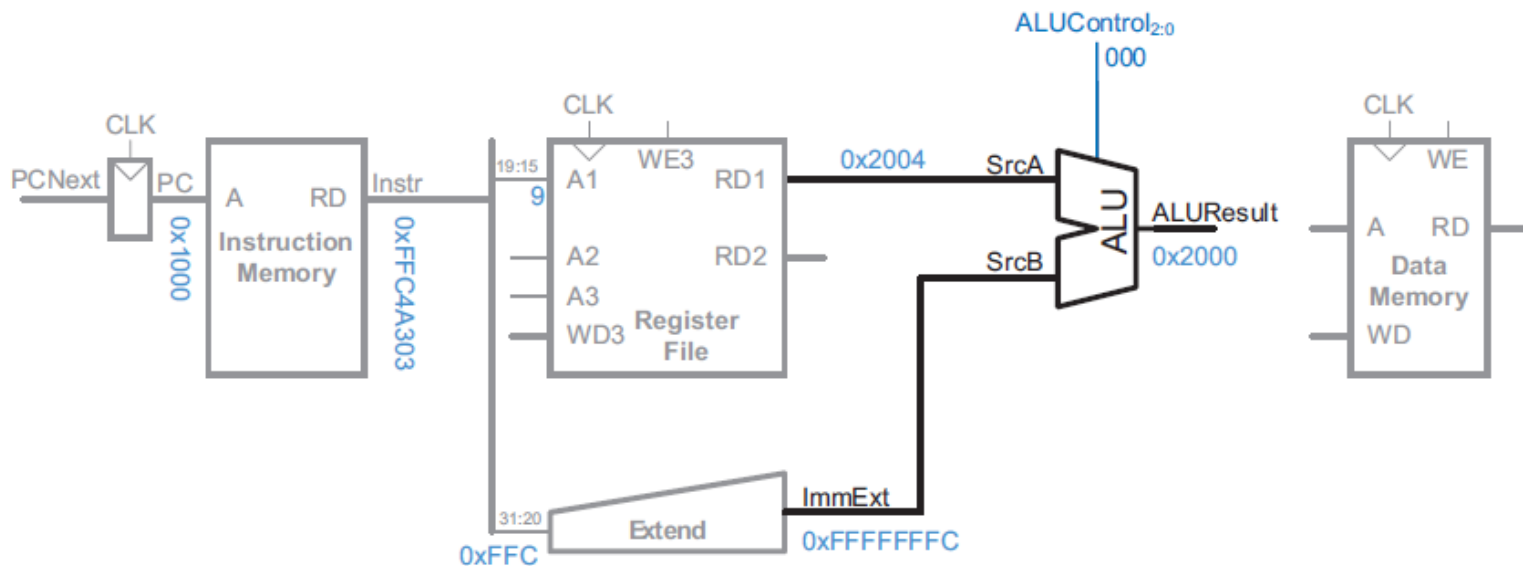
(c) extend the immediate operant (signed)



Address	Instruction	Type	Fields	Machine Language
0x1000	L7: lw x6, -4(x9)	I	<div> <div>imm_{11:0}</div> <div>111111111100</div> </div> <div> <div>rs1</div> <div>01001</div> </div> <div> <div>f3</div> <div>010</div> </div> <div> <div>rd</div> <div>00110</div> </div>	<div> <div>op</div> <div>0000011</div> </div> <div>FFC4A303</div>

Trace instruction execution

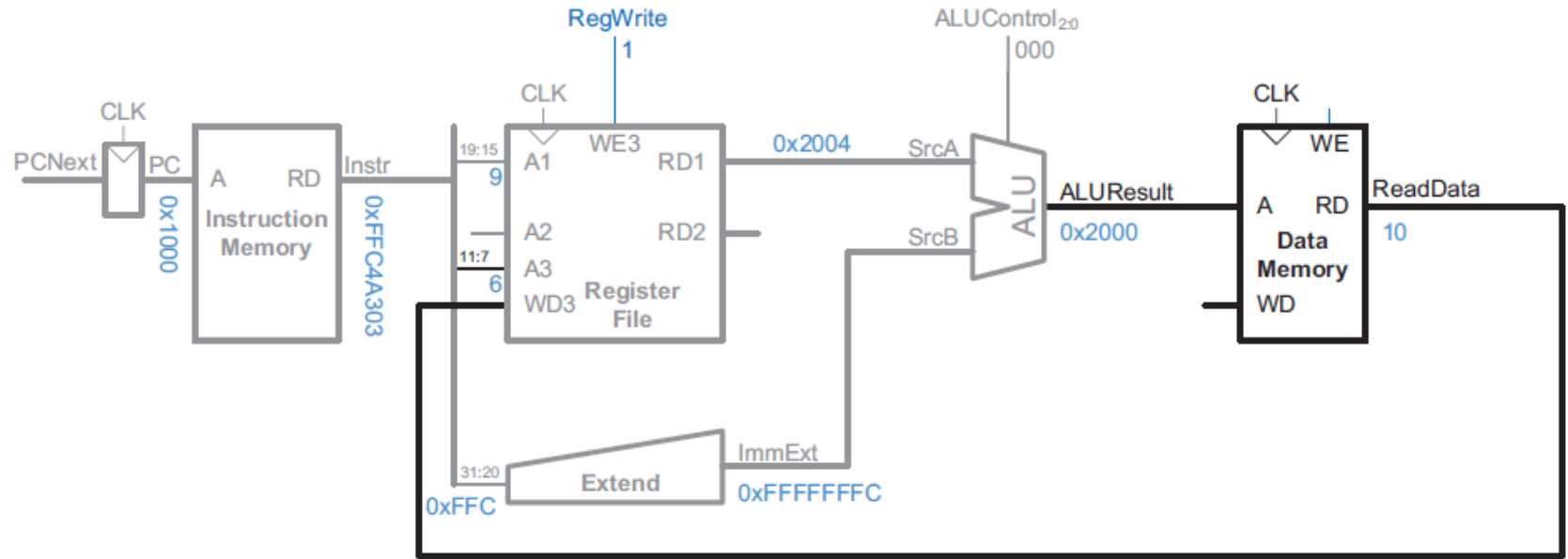
(d) compute memory address



Address	Instruction	Type	Fields			Machine Language	
0x1000	L7: lw x6, -4(x9)	I	imm _{11:0}	rs1	f3 rd	op	FFC4A303
			111111111100	01001	010 00110	0000011	

Trace instruction execution

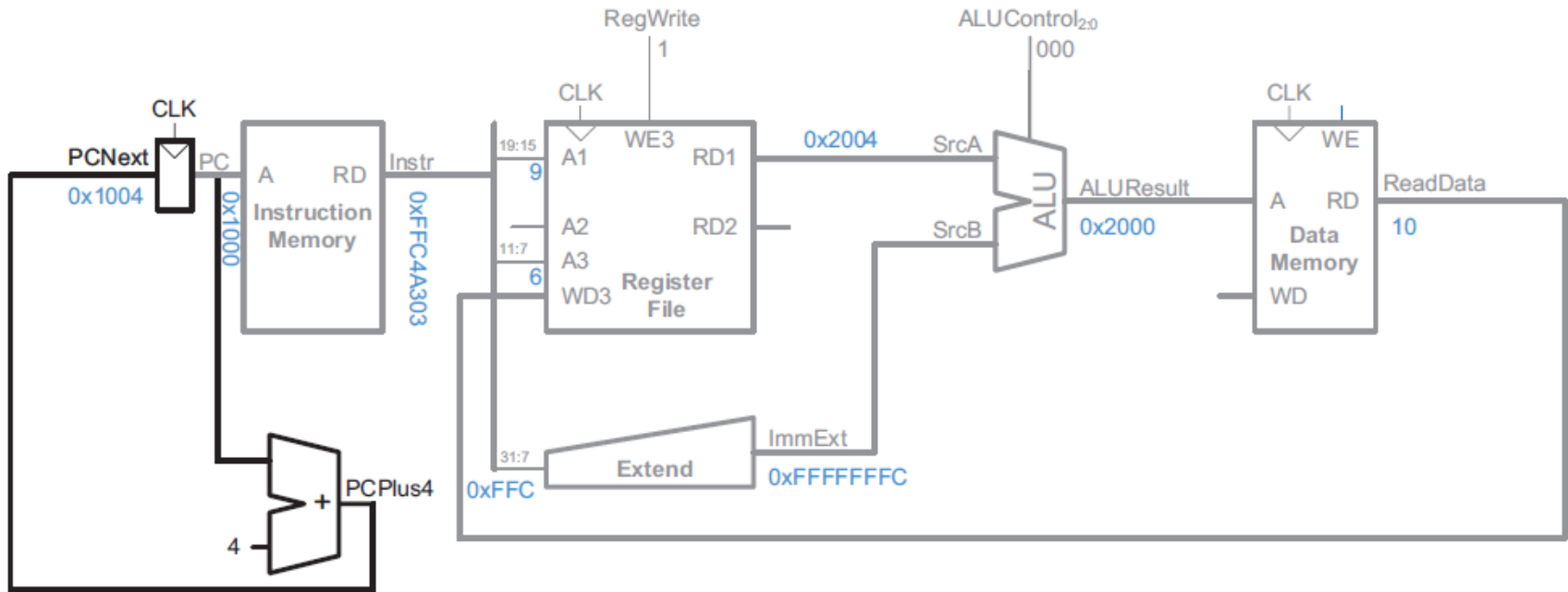
(e) read memory and write result to register file



Address	Instruction	Type	Fields			Machine Language	
			imm _{11:0}	rs1	f3	rd	op
0x1000	L7: lw x6, -4(x9)	I	111111111100	01001	010	00110	0000011
							FFC4A303

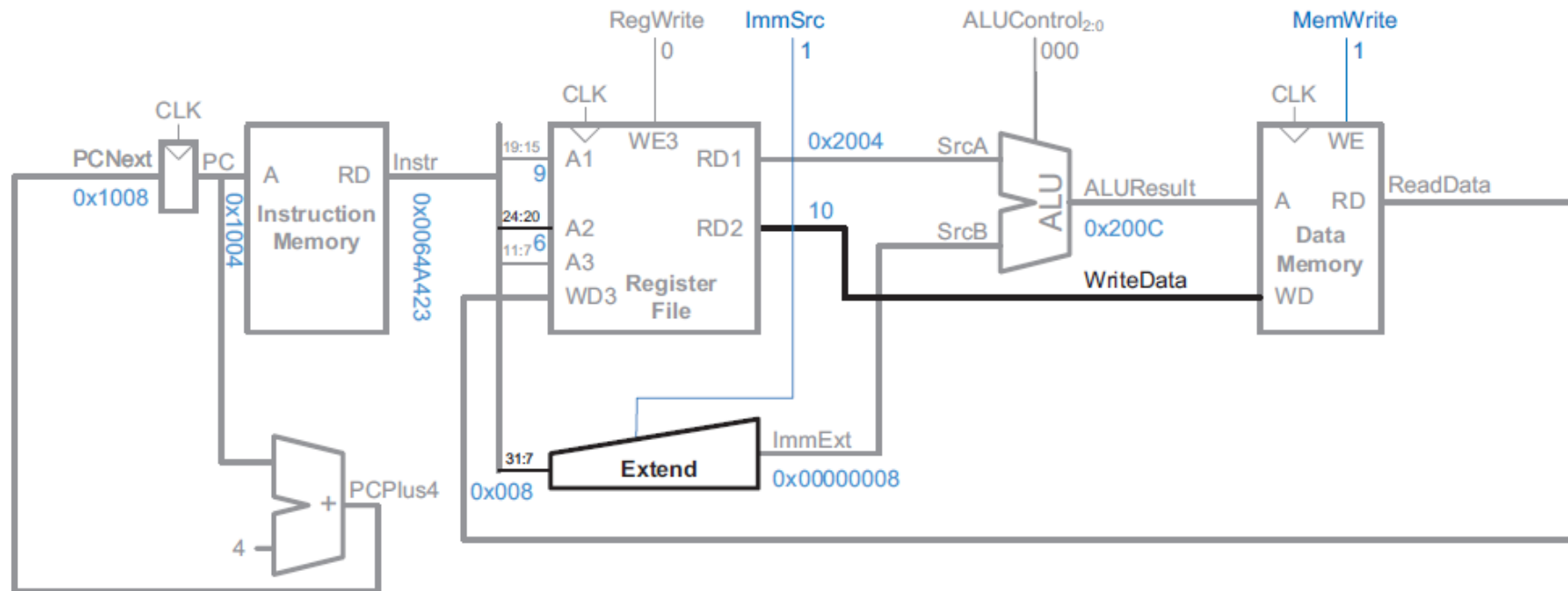
Trace instruction execution

(f) increment program counter



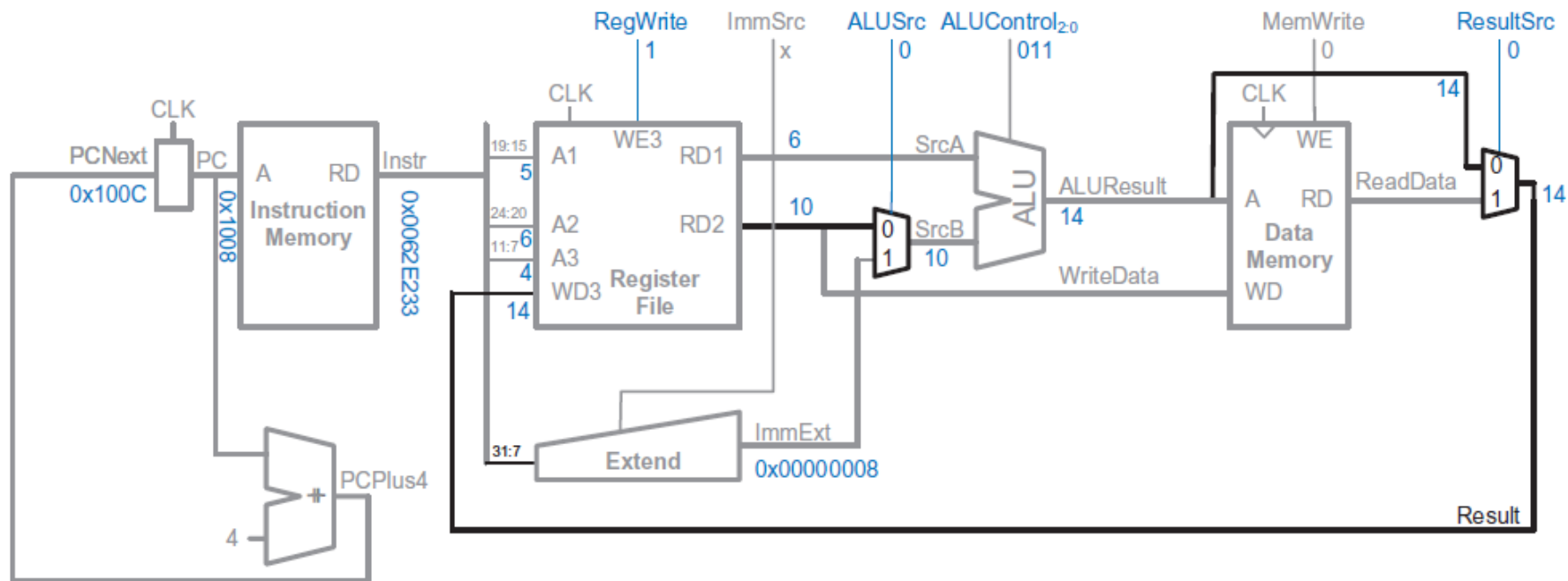
Address	Instruction	Type	Fields			Machine Language
0x1000	L7: lw x6, -4(x9)	I	imm _{11:0}	rs1	f3 rd op	FFC4A303
			111111111100	01001	010 00110 0000011	

Write data to memory



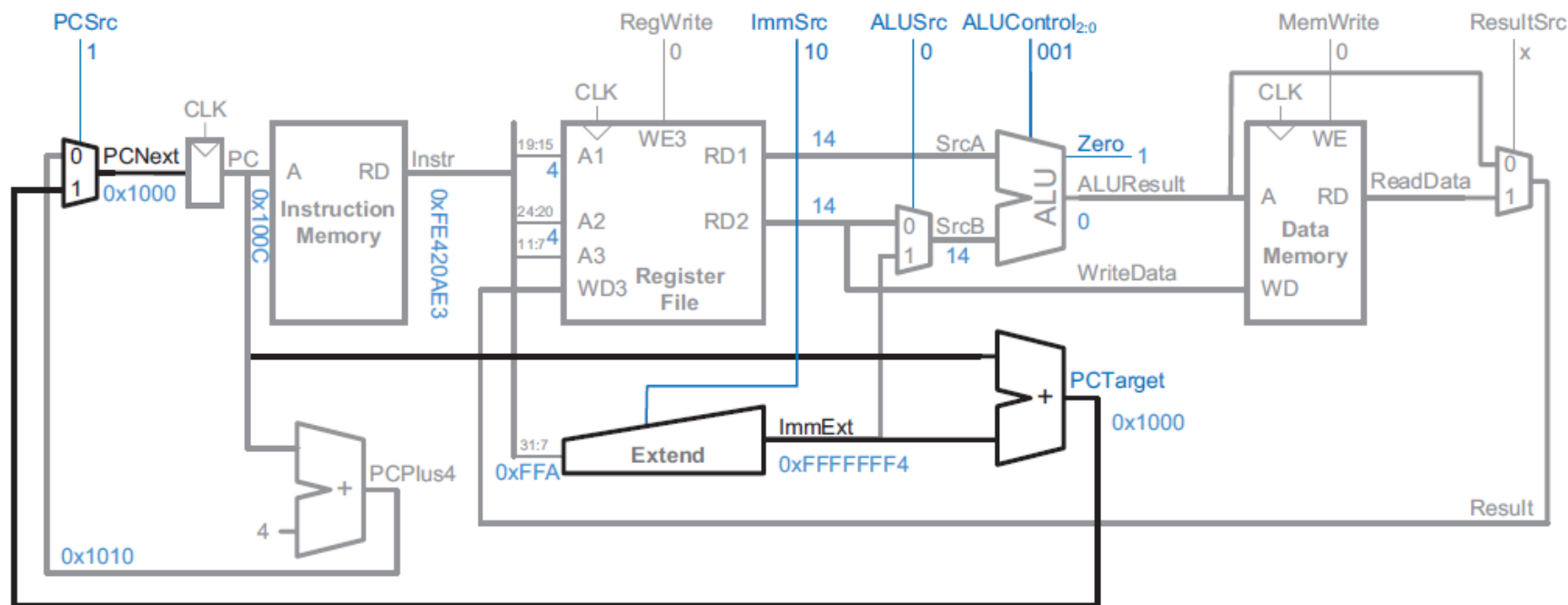
Address	Instruction	Type	Fields					Machine Language	
0x1004	sw x6, 8(x9)	S	imm _{11:5}	rs2	rs1	f3	imm _{4:0}	op	0064A423
			0000000	00110	01001	010	01000	0100011	

Binary (logic) operation



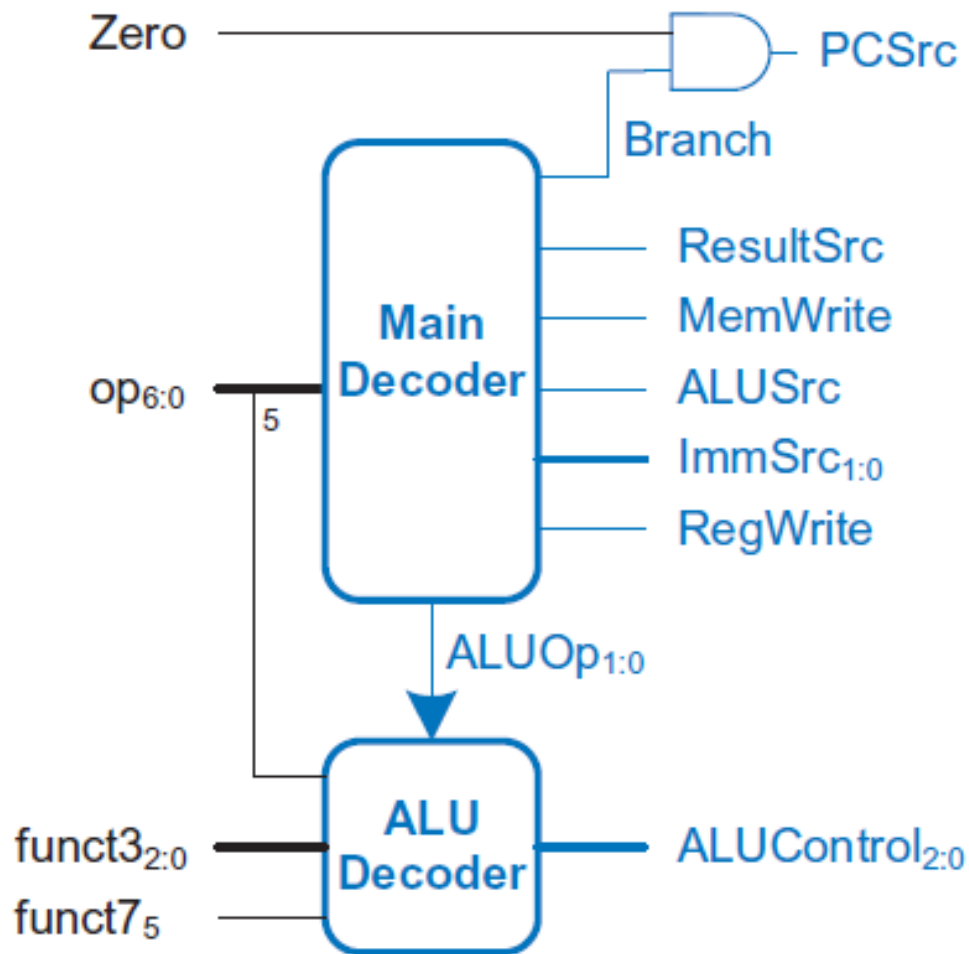
Address	Instruction	Type	Fields					Machine Language	
0x1008	or x4, x5, x6	R	funct7	rs2	rs1	f3	rd	op	0062E233
			0000000	00110	00101	110	00100	0110011	

Branching instruction



Address	Instruction	Type	Fields							Machine Language
0x100C	beq x4, x4, L7	B	imm _{12,10:5}	rs2	rs1	f3	imm _{4:1,11}	op		
			1111111	00100	00100	000	10101	1100011	FE420AE3	

Control unit design



Decoder tables

Instruction	Op	RegWrite	ImmSrc	ALUSrc	MemWrite	ResultSrc	Branch	ALUOp
lw	0000011	1	00	1	0	1	0	00
sw	0100011	0	01	1	1	x	0	00
R-type	0110011	1	xx	0	0	0	0	10
beq	1100011	0	10	0	0	x	1	01

ALU Decoder truth table

ALUOp	funct3	{op ₅ , funct ₇ }	ALUControl	Instruction
00	x	x	000 (add)	lw, sw
01	x	x	001 (subtract)	beq
10	000	00, 01, 10	000 (add)	add
	000	11	001 (subtract)	sub
	010	x	101 (set less than)	slt
	110	x	011 (or)	or
	111	x	010 (and)	and

Control and dataflow, binary operation

