Table 3: Main Controller Outputs

Instruction	**opcode**	**rf_we**	**sel_ext**	**sel_alu_src_b**	**dmem_we**	**sel_result**	**branch**	**jump**
R-type	0110011	1	xxx	0	0	00	0	0
I-type (arith)	0010011	1	001	1	0	00	0	0
lw	0000011	1	001	1	0	01	0	0
sw	0100011	0	010	1	1	xx	0	0
beq	1100011	0	011	0	0	xx	1	0
jal	1101111	1	100	x	0	10	0	1
lui	0110111	1	xxx	x	0	11	0	0

Control Signal Descriptions:

- **rf_we**: Register file write enable (1 = write, 0 = no write)
- **sel_ext**: Sign extender select (001=I-type, 010=S-type, 011=B-type, 100=J-type)
- **sel_alu_src_b**: ALU source B select (0=register, 1=immediate)
- **dmem_we**: Data memory write enable (1 = write, 0 = no write)
- **sel_result**: Result mux select (00=ALU, 01=Memory, 10=PC+4, 11=Immediate)
- **branch**: Branch signal (1 for beq)
- **jump**: Jump signal (1 for jal)

Note: PC source is computed as: `pc_src = (branch & alu_zero) | jump`

Table 4: ALU Decoder Outputs

Instruction	**opcode**	**funct3**	**funct7[5]**	**alu_control**	**Operation**
ADD	0110011	000	0	0000	A + B
SUB	0110011	000	1	0001	A - B
XOR	0110011	100	x	0010	A XOR B
OR	0110011	110	x	0011	A OR B
AND	0110011	111	x	0100	A AND B
SLL	0110011	001	x	0101	A << B
SRL	0110011	101	0	0110	A >> B (logical)
SRA	0110011	101	1	0111	A >> B (arithmetic)
SLT	0110011	010	x	1000	A < B (signed)
SLTU	0110011	011	x	1001	A < B (unsigned)
ADDI	0010011	000	x	0000	A + B
XORI	0010011	100	x	0010	A XOR B
ORI	0010011	110	x	0011	A OR B
ANDI	0010011	111	x	0100	A AND B
SLLI	0010011	001	x	0101	A << B
SRLI	0010011	101	0	0110	A >> B (logical)
SRAI	0010011	101	1	0111	A >> B (arithmetic)
SLTI	0010011	010	x	1000	A < B (signed)
SLTIU	0010011	011	x	1001	A < B (unsigned)
LW	0000011	010	x	0000	A + B
SW	0100011	010	x	0000	A + B
BEQ	1100011	000	x	0001	A - B

ALU Control Encoding:

0000=ADD, 0001=SUB, 0010=XOR, 0011=OR, 0100=AND, 0101=SLL, 0110=SRL, 0111=SRA, 1000=SLT, 1001=SLTU

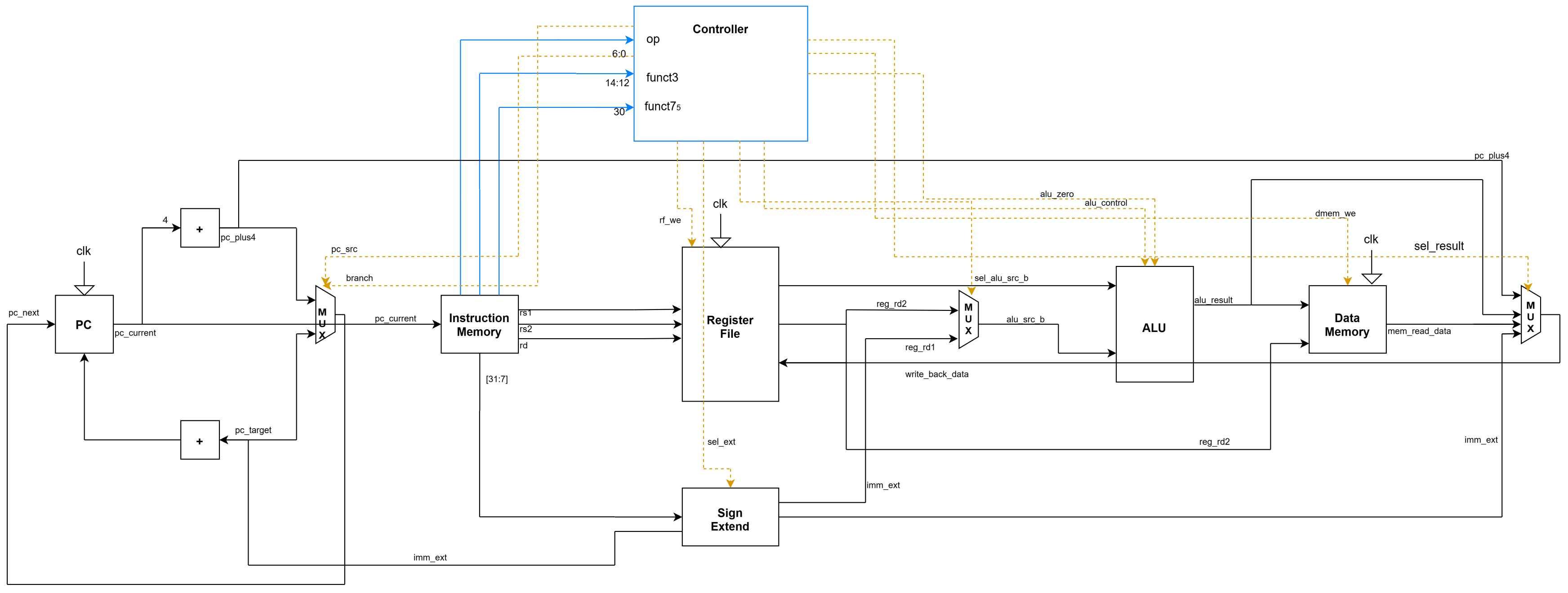

Figure 1: RISC-V Single-Cycle Processor Architecture