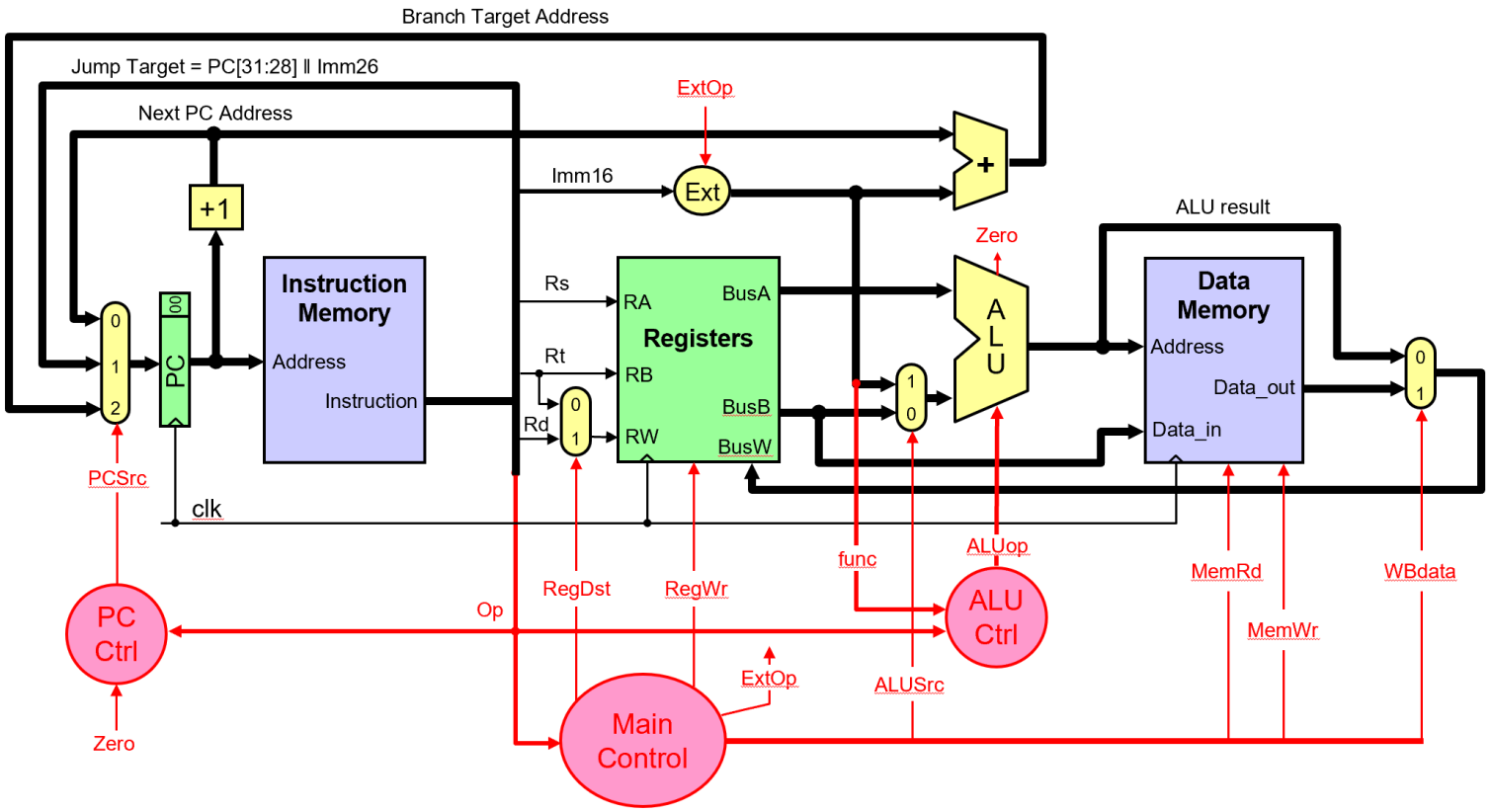
Computer Architecture

Practice - Single cycle processor design

Extend the single-cycle data-path shown below to support the following instructions. Add any necessary functional unites and control signals. Also, redesign the truth tables below (next page) and drive the new signals expressions.

1. **Lwr $rd, $rs, $rt** # Reg[rd] <— Mem[4*rs + rt]
2. **jalr r** # $ra <— PC+4
   # PC <— $r
3. **Cas** # Reg[rd] <— Max(Reg[rs] , Reg[rt])



Op	RegDst	RegWr	ExtOp	ALUSrc	MemRd	MemWr	WBdata
R-type	1 = Rd	1	X	0 = BusB	0	0	0 = ALU
ADDI	0 = Rt	1	1 = sign	1 = Imm	0	0	0 = ALU
SLTI	0 = Rt	1	1 = sign	1 = Imm	0	0	0 = ALU
ANDI	0 = Rt	1	0 = zero	1 = Imm	0	0	0 = ALU
ORI	0 = Rt	1	0 = zero	1 = Imm	0	0	0 = ALU
XORI	0 = Rt	1	0 = zero	1 = Imm	0	0	0 = ALU
LW	0 = Rt	1	1 = sign	1 = Imm	1	0	1 = Mem
SW	X	0	1 = sign	1 = Imm	0	1	X
BEQ	X	0	1 = sign	0 = BusB	0	0	X
BNE	X	0	1 = sign	0 = BusB	0	0	X
J	X	0	X	X	0	0	X

Op	funct	ALUOp	4-bit Coding
R-type	AND	AND	0001
R-type	OR	OR	0010
R-type	XOR	XOR	0011
R-type	ADD	ADD	0100
R-type	SUB	SUB	0101
R-type	SLT	SLT	0110
ADDI	X	ADD	0100
SLTI	X	SLT	0110
ANDI	X	AND	0001
ORI	X	OR	0010
XORI	X	XOR	0011
LW	X	ADD	0100
SW	X	ADD	0100
BEQ	X	SUB	0101
BNE	X	SUB	0101
J	X	X	X

Op	Zero flag	PCSrc
R-type	X	0 = Increment PC
J	X	1 = Jump Target Address
BEQ	0	0 = Increment PC
BEQ	1	2 = Branch Target Address
BNE	0	2 = Branch Target Address
BNE	1	0 = Increment PC
Other than Jump or Branch	X	0 = Increment PC