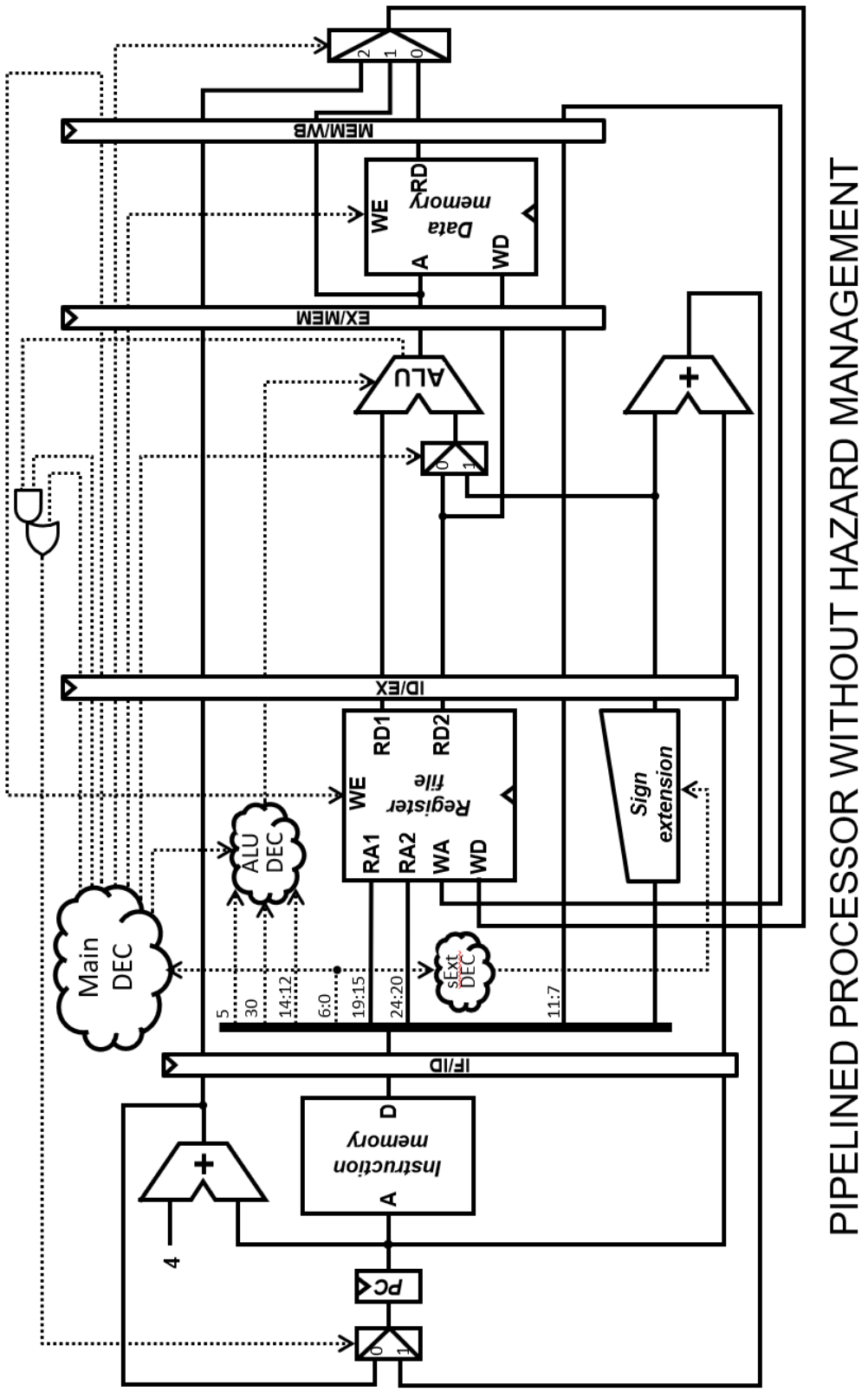
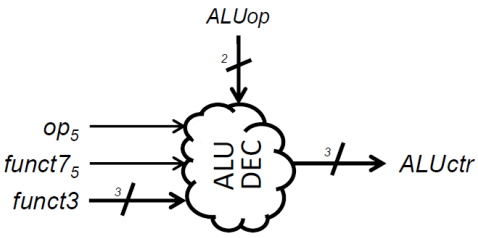
Facultad de Informática
Universidad Complutense de Madrid

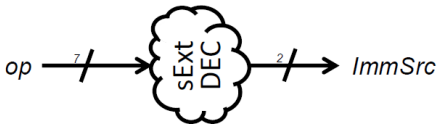
INTRODUCTION TO COMPUTERS II

PIPELINED MICROARCHITECTURE

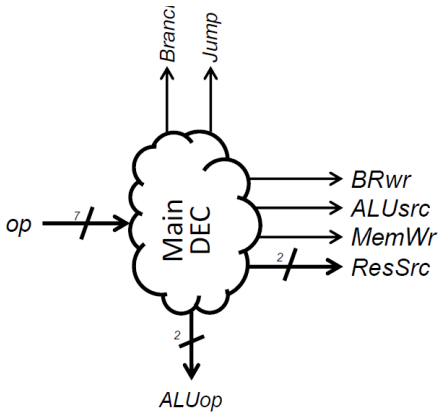




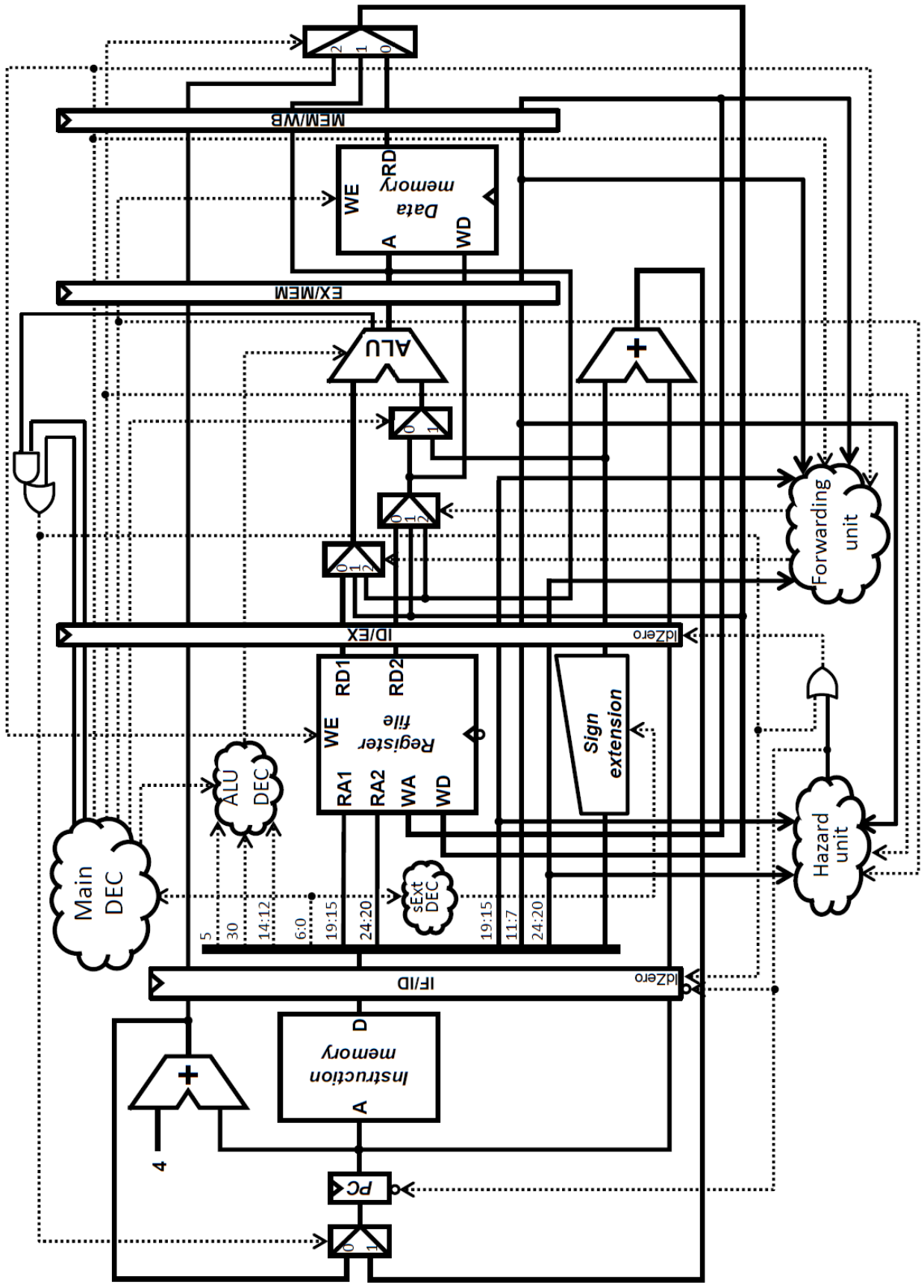
ALUop	op_5	$funct7_5$	funct3	ALUctr
00 (add)	X	X	XXX	000 (A + B)
01 (subtract)	X	X	XXX	001 (A – B)
10 (operate)	0	X	000 (addi)	000 (A + B)
10 (operate)	1	0	000 (add)	000 (A + B)
10 (operate)	1	1	000 (sub)	001 (A – B)
10 (operate)	X	X	010 (slt/slti)	101 (A < B)
10 (operate)	X	X	110 (or/ori)	011 (A | B)
10 (operate)	X	X	111 (and/andi)	010 (A & B)

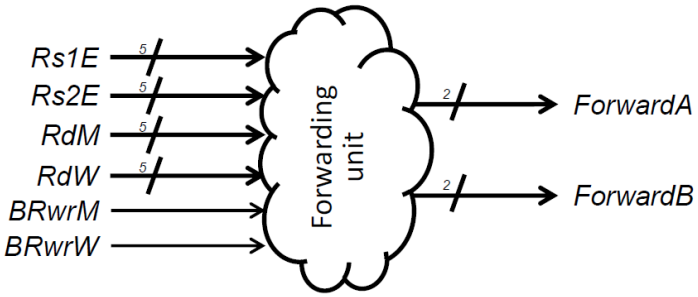


Op	ImmSrc
0000011 (lw)	00 (I-type)
0100011 (sw)	01 (S-type)
0010011 (I-type)	00 (I-type)
0110011 (R-type)	–
1100011 (beq)	10 (B-type)
1101111 (jal)	11 (J-type)



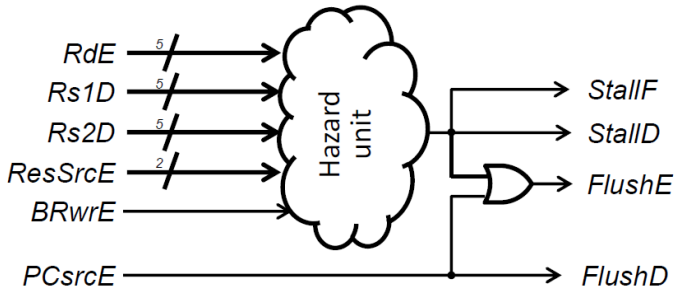
op	Branch	Jump	BRwr	ALUsrc	ALUop	MemWr	ResSrc
0000011 (lw)	0	0	1	1	00 (add)	0	00
0100011 (sw)	0	0	0	1	00 (add)	1	–
0010011 (I-type)	0	0	1	1	10 (operate)	0	01
0110011 (R-type)	0	0	1	0	10 (operate)	0	01
1100011 (beq)	1	0	0	0	01 (subtract)	0	–
1101111 (jal)	0	1	1	–	–	0	10


PIPELINED PROCESSOR WITH HAZARD MANAGEMENT
MEM/WB
Data memory
WE
RD
WD
A
EX/MEM
ALU
Forwarding unit
ID/EX
idZero
Register file
RD1
RD2
WE
RA1
RA2
WA
WD
Sign extension
ALU DEC
Main DEC
sExt DEC
Hazard unit
IF/ID
idZero
Instruction memory
D
A
PC
4
5
30
14:12
6:0
19:15
24:20
19:15
11:7
24:20



if ((Rs1E ≠ 0) & BRwrM & (Rs1E = RdM)) *then* (ForwardA ← $10^{\text{(forwarding MEM)}}$)
elsif((Rs1E ≠ 0) & BRwrW & (Rs1E = RdW)) *then* (ForwardA ← $01^{\text{(forwarding WB)}}$)
else (ForwardA ← $00^{\text{(no forwarding)}}$)

if ((Rs2E ≠ 0) & BRwrM & (Rs2E = RdM)) *then* (ForwardB ← $10^{\text{(forwarding MEM)}}$)
elsif((Rs2E ≠ 0) & BRwrW & (Rs2E = RdW)) *then* (ForwardB ← $01^{\text{(forwarding WB)}}$)
else (ForwardB ← $00^{\text{(no forwarding)}}$)



if ((ResSrcE = 0) & BRwrE & ((Rs1D = RdE) | (Rs2D = RdE))) *then* (lwStall ← 1) (stall pipeline)
else (lwStall ← 0) (don't stall pipeline)
StallF = StallD = lwStall
FlushD = PCsrcE
FlushE = lwStall | PCsrcE