Instruction	Description
STOP	stop execution
B	branch
BL	branch with link
BR	branch register
CBZ	compare and branch if zero
CBNZ	compare and branch if not zero
STURB	store byte
LDURB	load byte
STURH	store half word
LDURB	load half word
STURW	store word
LDURSW	load signed word
STUR	store register
LDUR	load register
STXR	store exclusive register
LDXR	load exclusive register

Table E.1: LEGv8 Instructions

Instruction	Description
ADD	add
SUB	subtract
AND	and
ORR	or
EOR	exclusive or
ADDI	add immediate
SUBI	subtract immediate
ANDI	and immediate
ORRI	or immediate
EORI	exclusive or immediate
ADDS	add & set flags
SUBS	subtract & set flags
ANDS	and & set flags
ADDIS	add immediate & set flags
SUBIS	subtract immediate & set flags
ANDIS	and immediate & set flags
LSR	logical shift right
LSL	logical shift left
MOVZ	move wide with zero
MOVK	move wide with keep
MUL	multiply
UMULH	unsigned multiply high
SMULH	signed multiply high
UDIV	unsigned divide
SDIV	signed divide

Table E.2: LEGv8 Instructions (continued)

inst	meaning after `SUBS`
`B.EQ`	equal
`B.NE`	not equal
Useful for Signed Computation	
`B.LT`	less than
`B.GE`	greater than or equal to
`B.LE`	less than or equal to
`B.GT`	greater than
`B.MI`	negative
`B.PL`	non-negative
`B.VS`	overflow
`B.VC`	no overflow
Useful for Unsigned Computation	
`B.LO`	lower than
`B.HS`	higher than or same
`B.LS`	lower than or same
`B.HI`	higher than
`B.CS`	branch if carry set
`B.CC`	branch if carry clear

Table E.3: Advanced Branching instructions.

R	op	Rm	shamt	Rn	Rd
	11 bits	5 bits	6 bits	5 bits	5 bits
I	op	immediate		Rn	Rd
	10 bits	12 bits		5 bits	5 bits
D	op	offset	op2	Rn	Rt
	11 bits	9 bits	2 bits	5 bits	5 bits
B	op	branch offset			
	6 bits	26 bits			
CB	op	branch offset			Rt/cond
	8 bits	19 bits			5 bits

Table E.4: Instruction formats.

CMP	X1, X2	=	SUBS	XZR, X1, X2
CMPI	X1, #imm	=	SUBIS	XZR, X1, #imm
LDA	X1, #addr	=	MOVZ	X1, #addr<63:48>, LSL #48
			MOVK	X1, #addr<47:32>, LSL #32
			MOVK	X1, #addr<31:16>, LSL #16
			MOVK	X1, #addr<15:0>, LSL #0

Table E.5: Psudo-instructions.

Name	Number	Usage
	0–7	Arguments and Return values
	8	Indirect result location (not used in the course)
	9–15	Temporary
	16–18	Reserved (not used in the course)
	19–26	Saved registers
`GP`	27	Global pointer
`SP`	28	Stack pointer
`FP`	29	Frame pointer
`LR`	30	Return address
`XZR`	31	Constant 0

Table E.6: Register designations.

31	30–23	22–0
sign	exponent	mantissa

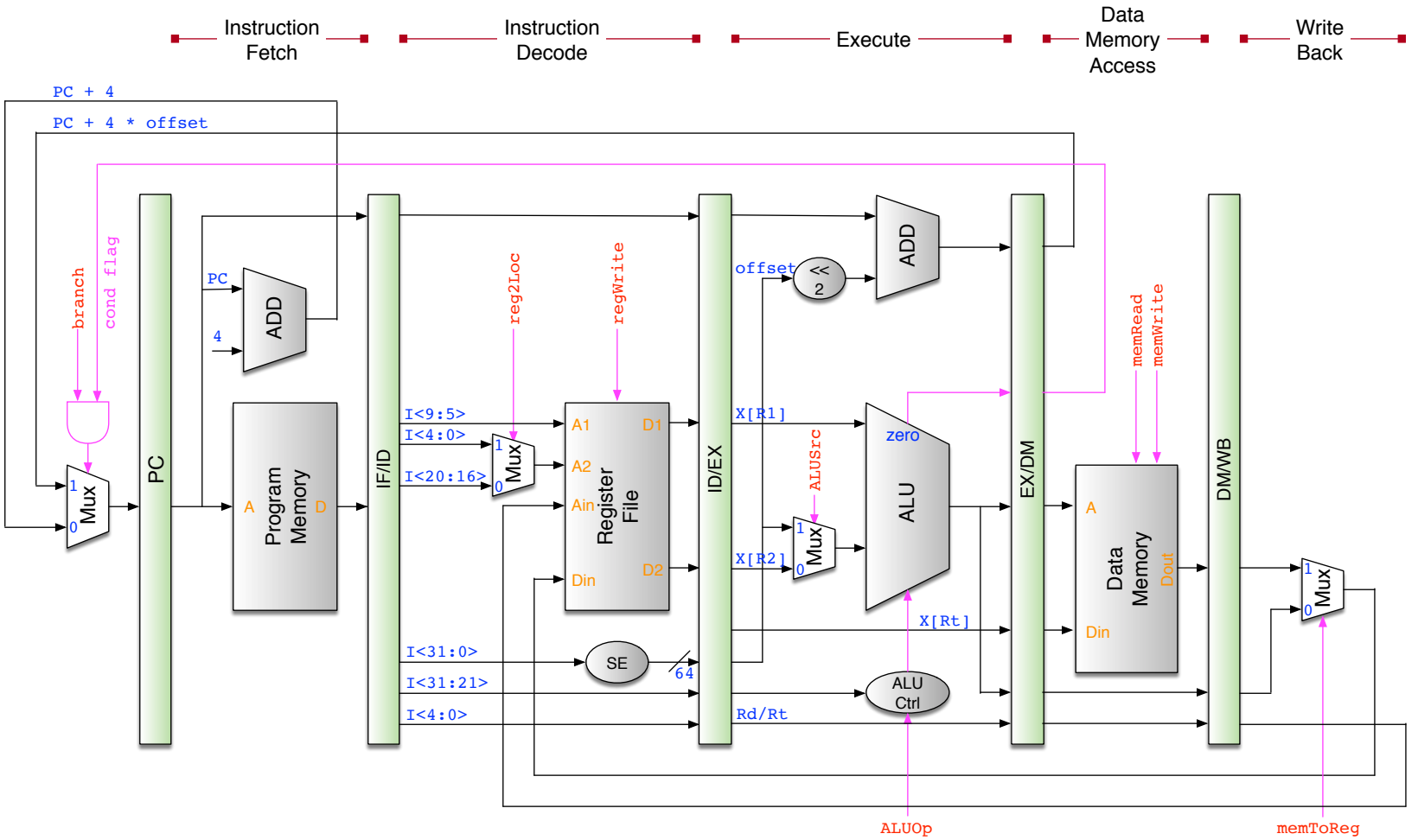
Table E.7: Floating point: Exponent is a *biased* integer (bias = 127)

63	62–52	51–0
sign	exponent	mantissa

Table E.8: Double Precision: Exponent is a *biased* integer (bias = 1023)

ALUOp	Action
00	add
01	check second input = 0
10	follow *opcode*

$$1\text{GB} = 2^{30} \text{ Bytes}$$
$$1\text{MB} = 2^{20} \text{ Bytes}$$
$$1\text{KB} = 2^{10} \text{ Bytes.}$$


Instruction Fetch
Instruction Decode
Execute
Data Memory Access
Write Back
PC + 4
PC + 4 * offset
branch
cond flag
PC
ADD
4
Mux
1
0
PC
Program Memory
A
D
IF/ID
I<9:5>
I<4:0>
I<20:16>
reg2Loc
regWrite
A1
A2
Ain
Din
D1
D2
Register File
I<31:0>
I<31:21>
I<4:0>
SE
64
ID/EX
offset
<< 2
ADD
X[R1]
X[R2]
ALUSrc
zero
ALU
X[Rt]
ALU Ctrl
Rd/Rt
EX/DM
memRead
memWrite
Data Memory
Dout
DM/WB
ALUOp
memToReg