

up2

Ashley J. Robinson

November 26, 2017



1

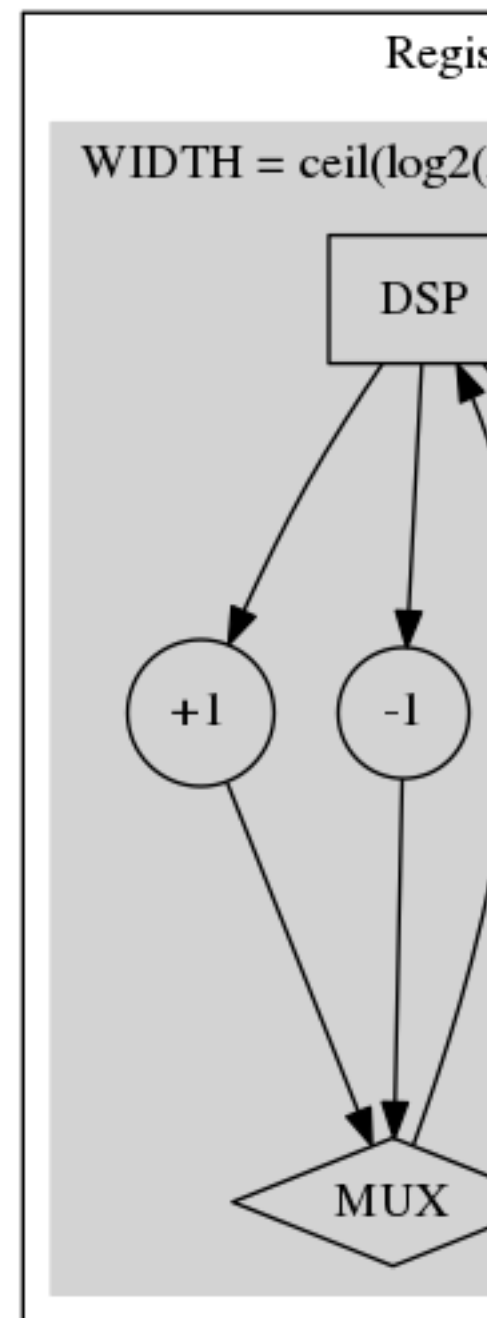
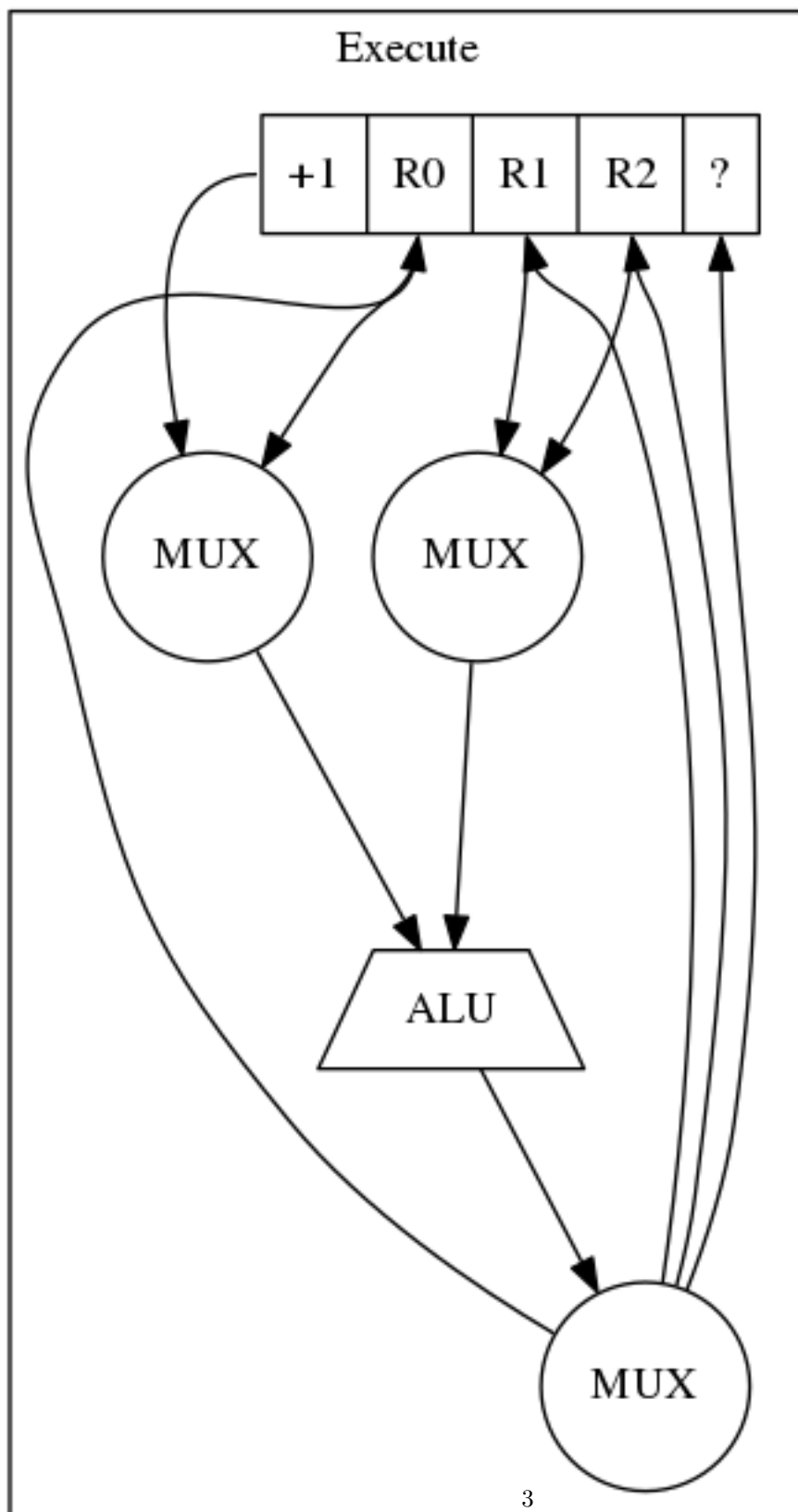


Table 1: My caption

Opcode	Opcode				
Hex	Binary	Mnemonic	Use MX	Description	Flags set
0x0	0000	ADD	Yes	$R_0, R_1, R_2, ?_i = R_1, R_2_i + i+1, R_0_i$	Z
0x1	0001	SUB	Yes	$R_0, R_1, R_2, ?_i = R_1, R_2_i - i+1, R_0_i$	Z
0x2	0010	OR	Yes	$R_0, R_1, R_2, ?_i = R_1, R_2_i \text{ OR } i+1, R_0_i$	Z
0x3	0011	NOR	Yes	$R_0, R_1, R_2, ?_i = R_1, R_2_i \text{ NOR } i+1, R_0_i$	Z
0x4	0100	XOR	Yes	$R_0, R_1, R_2, ?_i = R_1, R_2_i \text{ XOR } i+1, R_0_i$	Z
0x5	0101	NAND	Yes	$R_0, R_1, R_2, ?_i = R_1, R_2_i \text{ NAND } i+1, R_0_i$	Z
0x6	0110	LSL	Yes	$R_0, R_1, R_2, ?_i = R_1, R_2_i \ll 1$	Z
0x7	0111	LSR	Yes	$R_0, R_1, R_2, ?_i = R_1, R_2_i \gg 1$	Z
0x8	1000	BNE	No	if( $Z \neq 1$ ) PC += W	
0x9	1001	BE	No	if( $Z == 1$ ) PC += W	
0xA	1010	JMP	No	PC = CS[PC]	
0xB	1011	INT	No	Enable ints	
0xC	1100	DDS	No	DSP -= 1	
0xD	1101	IDS	No	DSP += 1	
0xE	1110	DPS	No	PSP -= 1	
0xF	1111	IPS	No	PSP += 1	

## 2

### 2.1 ADD

### 2.2 SUB

### 2.3 OR

### 2.4 NOR

### 2.5 XOR

### 2.6 NAND

### 2.7 LSL

Logical Shift Left

### 2.8 LSR

Logical Shift Right

### 2.9 BNE

Branch Not Equals

### 2.10 BE

Branch Equals

## **2.11 JMP**

Jump

## **2.12 INT**

Enable/Disable interrupts

## **2.13 DDS**

Decrement Data Stack Pointer

## **2.14 IDS**

Increment Data Stack Pointer

## **2.15 DPS**

Decrement Program Counter Stack Pointer

## **2.16 IPS**

Increment Program Counter Stack Pointer

# **3 Programming Tips**

## **3.1 Zero a register**

Objective: Set R1 to zero

LSL,R1,R1 LSL,R1,R1 LSL,R1,R1 LSL,R1,R1

## **3.2 Load Immediate**

Objective: Load 0xA(1010) in to R2

OR,R1,R1,+1 LSL LSL OR,R1,R1,+1 LSL

## **3.3 Swapping registers**

Objective: Swap R0 and R1

XOR,R0,R1,R0 XOR,R1,R1,R0 XOR,R0,R1,R0