Darrian Johnson
ELEC 5200 – Project Part 1
2/10/2021

RISC – it all

Type	Name	Mnemonic	Opcode	Description/Function
R	Add	plus	0000	Satisfies '+' operator
R	Subtract	min	0001	Satisfies '-' operator
R	AND	and	0010	Satisfies bitwise 'and(&)' operator
R	OR	or	0011	Satisfies bitwise 'or(|)' operator
I	Load Word	ldw	0100	Fetches a word from memory into registers
I	Store Word	stw	0101	Stores word into memory from registers
I	Add Immediate	plusi	0110	Adds constant values to register values
I	Load Upper Immediate	lui	0111	Loads upper byte of word with value from memory or constant
B	Branch Equal	beq	1000	Satisfies '==' operator
B	Branch Not Equal	bne	1001	Satisfies '!=' operator
B	Branch Greater Than	bgt	1010	Satisfies '>' operator
B	Branch Less Than	blt	1011	Satisfies '<' operator
B	Branch Greater Than or Equal	bge	1100	Satisfies '>=' operator
B	Branch Less Than or Equal	blte	1101	Satisfies '<=' operator
J	Jump	jmp	1110	Jumps to subroutine
N/A	HALT	stop	1111	Ends program

Register Types

R-Type				
	rs2	**rs1**	**rd**	**opcode**
bits	[15:12]	[11:8]	[7:4]	[3:0]

I-Type				
	imm	**rs1**	**rd**	**opcode**
bits	[15:12]	[11:8]	[7:4]	[3:0]

B-Type				
	imm	**rs1**	**rd**	**opcode**
bits	[15:12]	[11:8]	[7:4]	[3:0]

J-Type				
	imm	**rs1**	**rd**	**opcode**
bits	[15:12]	[11:8]	[7:4]	[3:0]

User Programmable Registers

Register	**Binary Code**	**Name**	**Function**
0	0000	zero	Constant value 0
1	0001	$s0	Storage Registers
2	0010	$s1	
3	0011	$s2	
4	0100	$s3	
5	0101	$s4	
6	0110	$s5	
7	0111	$s6	
8	1000	$t0	Temp Registers
9	1001	$t1	
10	1010	$t2	
11	1011	$t3	
12	1100	$t4	
13	1101	$t5	
14	1110	$t6	
15	1111	$ret	Return Address

Non-User Programmable Register

Register	**Function**	**Bit Width**
PC	Program Counter	10-bits

Name	Mnemonic	Assembly Code	Machine Code
Add	plus	plus $s0,$s1,$s2	0011 0010 0001 0000
Subtract	min	min $s0,$s1,$s2	0011 0010 0001 0001
AND	and	and $s0,$s1,$s2	0011 0010 0001 0010
OR	or	or $s0,$s1,$s2	0011 0010 0001 0011
Load Word	ldw	ldw $s4,7(zero)	0111 0000 0101 0100
Store Word	stw	stw $s4,6($s5)	0110 0101 0110 0101
Add Immediate	plusi	plusi $s0,$s1,5	0101 0010 0001 0110
Load Upper Immediate	lui	lui $t0,127($ret)	0111 1111 1000 0111
Branch Equal	beq	beq $t4,$t5,1	0001 1101 1100 1000
Branch Not Equal	bne	bne $t4,$t5,3	0011 1101 1100 1001
Branch Greater Than	bgt	bgt $t4,$t5,5	0101 1101 1100 1010
Branch Less Than	blt	blt $t4,$t5,7	0111 1101 1100 1011
Branch Greater Than or Equal	bge	bge $t4,$t5,-1	1001 1101 1100 1100
Branch Less Than or Equal	blte	blte $t4,$t5,-3	1011 1101 1100 1101
Jump	jmp	jmp $ret,-5($t0)	1101 1000 1111 1110
HALT	stop	-	xxxx xxxx xxxx 1111

Name	Mnemonic	Assembly Code	C Code
Add	plus	plus $s0,$s1,$s2	a = b+c;
Subtract	min	min $s0,$s1,$s2	a = b-c;
AND	and	and $s0,$s1,$s2	a = b&c;
OR	or	or $s0,$s1,$s2	a = b|c;
Load Word	ldw	ldw $t1,7($t0) plusi $t1,$t1,5 stw $t1,0($t0)	0111 1111 1000 0111
Add Immediate	plusi		
Store Word	stw		
Load Upper Immediate	lui	lui $t0,127($ret)	(?)
Branch Equal	beq	beq $t4,$t5,1	if (a==b)…; else …;
Branch Not Equal	bne	bne $t4,$t5,3	if (a!=b)…; else…;
Branch Greater Than	bgt	bgt $t4,$t5,5	if(a>b)…; else…;
Branch Less Than	blt	blt $t4,$t5,7	if(a<b)…; else…;
Branch Greater Than or Equal	bge	bge $t4,$t5,-1	if(a>=b)…; else…;
Branch Less Than or Equal	blte	blte $t4,$t5,-3	if (a<=b)…; else…;

Jump	jmp	jmp $ret,-5($t0)	main(){
			subroutine();
			}
			int
			subroutine(void){
			…
			return();
			}

HALT	stop	(?)	exit();