CMSC132 Project Requirements
T-5L

Group Name: Tres Magagandang Marias

MAPPING OF THE OPCODE TO MACHINE CODE

INSTRUCTION	INSTRUCTION CODE	BINARY CODE
LOAD	LD	0000 0000
STORE	STR	0000 0001
SAVE	SV	0000 0010
INC	INC	0000 0011
DEC	DEC	0000 0100
ADD	ADD	0000 0101
SUB	SUB	0000 0110
MUL	MUL	0000 0111
DIV	DIV	0000 1000
CMP	CMP	0000 1001
AND	AND	0000 1010
OR	OR	0000 1011
NOT	NOT	0000 1100
XOR	XOR	0000 1101
JE	JE	0000 1110
JG	JG	0000 1111
JL	JL	0001 0000
JMP	JMP	0001 0001

MAPPING OF REGISTERS TO REGISTER NUMBER

REGISTER	REGISTER NUMBER
r0	0000 0000
r1	0000 0001
r2	0000 0010
r3	0000 0011
r4	0000 0100
r5	0000 0101
r6	0000 0110
r7	0000 0111
MEMORY ADDRESS REGISTER	
mar0	0000 1000
mar1	0000 1001
PC (Program Counter)	Address of next instruction
IR (Instruction Register)	Current Instruction
Flag	Zero flag and Sign flag

DEFINITION OF INSTRUCTIONS

Data Transfer Instructions

LOAD – loads a data from a specify memory address to the specified registers.
Syntax: LD <destination> <source>
source – memory address
destination – register

STORE – stores a data from the specified register to a memory location.
Syntax: STR <destination> <source>
source – register
destination – memory address

SAVE – store a value in a given register.
Syntax: SV <destination> <source>
source – immediate, another register
destination – register

Arithmetic Instructions

INC – increments the value in the given register and stores the incremented value in another register.
Syntax: INC <destination>
destination – register

DEC – decrements the value in the given register and stores the decremented value in another register.
Syntax: DEC <destination>
destination – register

ADD – adds the value of the first register to the second register and stores the result on the first register.
Syntax: ADD <destination> <source>
source – immediate, another register, memory address register
destination – register, memory address register

SUB – subtracts the value of the second register to the first register and stores the result on the first register.
Syntax: SUB <destination> <source>
source – immediate, another register, memory address register; acts as subtrahend
destination – register, memory address register; acts as minued and difference

MUL – multiplies the value of the first register to the second register and stores the result on the first register.
Syntax: MUL <destination> <source>
source – immediate, another register, memory address register
destination – register, memory address register

DIV – divides the value of the first register by the second register and stores the result on the first register.
Syntax: DIV <destination> <source>
source – immediate, another register, memory address register; acts as divisor
destination – register, memory address register; acts as dividend and quotient

Comparison Operation

CMP – stores in a specified register the value:
0000 0000 if the first register is equal to the second register
0000 0001 if the first register is greater than the second register
0000 0010 if the first register is less than the second register

Logic Instructions

AND – performs bitwise AND operation between the first and second register and stores the result in the first register.
Syntax: AND <destination> <source>
source – immediate, another register, memory address register
destination – register, memory address register

OR – performs bitwise OR operation between the first and second register and stores the result in the first register.
Syntax: OR <destination> <source>
source – immediate, another register, memory address register
destination – register, memory address register

*NOT – performs bitwise NOT operation of the second register and stores the result in the first register.
Syntax: NOT <destination> <source>
source – immediate, another register, memory address register
destination – register, memory address register

XOR – performs bitwise XOR operation between the first and second register and stores the result in the first register.
Syntax: XOR <destination> <source>
source – immediate, another register, memory address register
destination – register, memory address register

Program Flow Instructions

JE – if the value in the zero flag is equal to 1, the value of the PC is set to be equal to the specified memory address.
Syntax: JE <source>
source – memory address

JG – if both the value of the zero flag and the sign flag is equal to 0, the value of the PC is set to be equal to the specified memory address.
Syntax: JG <source>
source – memory address

JL – if the value in the sign flag is equal to 1, the value of the PC is set to be equal to the specified memory address.
Syntax: JL <source>
source – memory address

JMP – the value of the PC is set to be equal to the specified memory address.
Syntax: JMP <source>
source – memory address

Note: <source> and <destination> should not be both memory address registers.