TreadMill

COA Lab Project (2016-17)
The LNMIIT, Jaipur

Khushboo Baheti (15UCS168)
Smriti Jha (15UCS142)
Shreya Rajput (15UCS134)
Sakshi Goyal (15UCS116)

Introduction

- We are designing a **16-bit** Treadmill Machine.
- Includes functions that can measure and analyse procedures or the levels of an activity that can properly attribute performance factor.

  Example:

- **CaloriesBurnt**: It will tell the amount of calories burnt during a particular run for a particular time.
- **VO2Consumption**:which will tell the amount of oxygen consumed in a run.

Functions

1. VO2Consumption:Vo2 measurement represents the rate of oxygen consumption in that activity,which is, ***VO2(ml/(kg*min))***=[(0.2*speed(m/min))+(0.9*speed*% grade)+3.5
2. setGrade:This function is used to initialise the inclination of the treadmill floor in terms of tan theta taken as input.
3. MET: Metabolic Equivalent of Task, 1 met is the energy equivalent expended by an individual while seated at rest.(1MET=3.5ml/(kg*min))
   MET(mets)=VO2/3.5
4. CaloriesBurnt: This can be derived from the relation
   Calories burnt:Time spent running(min)*[MET value*3.5*weight of body (in kg)/ 200]
5. CalculateDistance:Given the speed(m/min) and the time (min),***Distance(m)***=speed/time.

6. SetSpeed:This function is used to initialise the speed of machine (in m/min).

7. Start:This function starts the treadmill by taking time of running (in minutes) as an argument. It starts the timer accordingly.

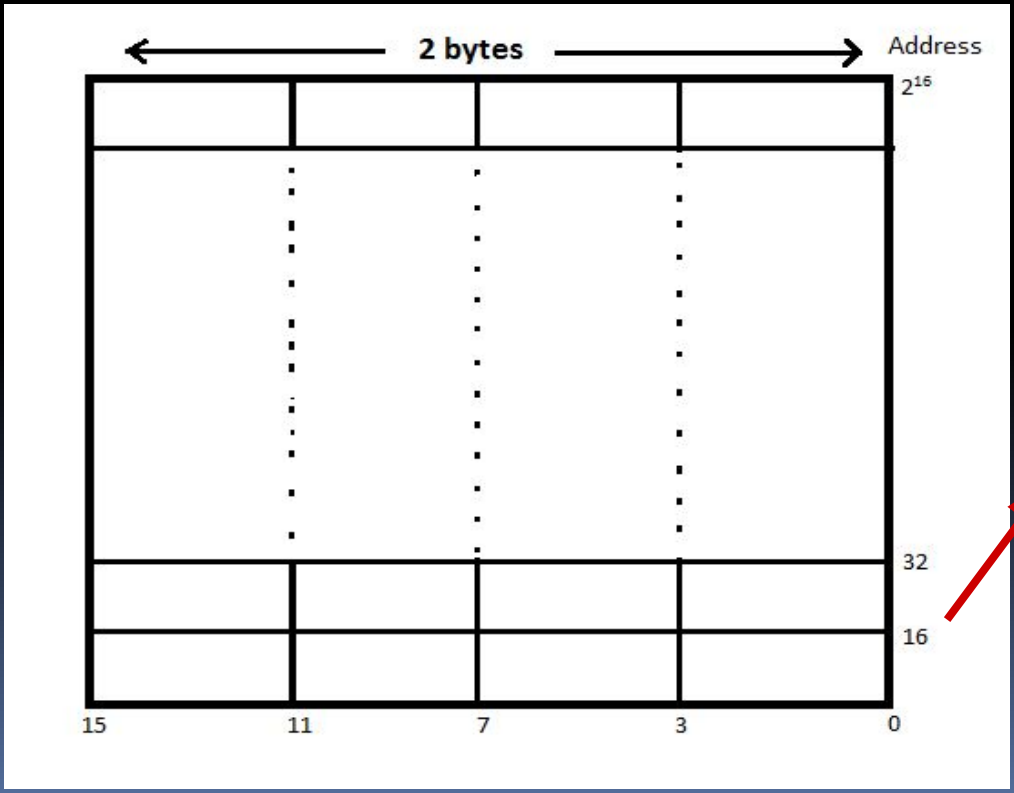
8. SpeedInc:This will increment the current speed by 1.

9. SpeedDec:This will decrement the current speed by 1.

10. SetWeight: This function will initialise the weight as the weight given by the user(in kgs).

Memory Model

- We are using Little Endian model,where word is made up of 16 bits(2 bytes).
- Our memory is word addressable memory.
- We are using an Aligned memory model.



Registers

- R0-R10 are **general purpose** registers which will be used to store key local variables and immediate results of calculations.

  R0 0000
  R1 0001
  . .
  R6 0101

- **Special Registers**:

GRD	0111
WGT	1000
TME	1001
SPD	1010
IN	1011
OUT	1100
PC	1101
IR	1110
FR	1111

- Special-purpose registers control various caches, memory, I/O devices and other hardware features.
- IN/OUT registers will be used to take input & display output.
- GRD(Grade),WGT(Weight),TME(Time),SPD(Speed),PC(Program Counter),IR(Instruction register),FR(Flag register) are special purpose registers.

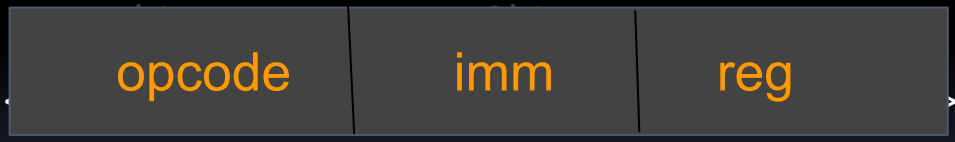
Instruction Format

- **2 - Address Instructions:**

<------8 bits ---><----4 bits-----><----- 4 bits------>

Example-	caloxygen	R0	R1
	caldist	R1	R2

- **Immediate Address Instruction:**



Example-	setspeed	imm	R1
	settime	imm	R1
	setweight	imm	R1
	setgrade	imm	R1

- **1- Address Instructions:**

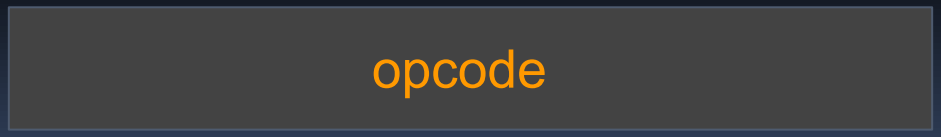
<---------12 bits-------><----------4 bits---------->

Example-	caloburnt	R1
	calmet	R1
	spdinc	R1
	spddec	R1

<-----------4bits------------><-------------12bits-------->

Example-	Jump	LABEL

- o-Address Instructions:



<----------------------16bits----------------------->

Example- HLT

In our instructions, second operand will be the destination register in which the result will be stored.

Instruction Design

4 bits:	setspeed	0000
	settime	0001
	setweight	0010
	setgrade	0011
	Jump	0100
8 bits:	caldist	01010000
	caloxygen	01010001
12 bits:	caloburnt	010100100000
	calmet	010100100001
	spdinc	010100100010
	spddec	010100100011
16 bits:	HLT	0101001001000000

- RISC due to faster clock speeds.
- Fixed length instructions because of easy decoding.

Instruction Types

- Data Movement:
  Register to Register= caldist
  Immediate to Register=setspeed
- Dyadic Instructions:

caldist	op1 op2	settime	op1 op2
caloxygen	op1 op2	setweight	op1 op2
setspeed	op1 op2	setgrade	op1 op2

- Monodic Instructions:

caloburnt	op1	spdinc	op1
calmet	op1	Jump	label

- Branching:

Jump	label

Data Types

We are using only Unsigned Integers .

Addressing Modes

- Immediate Addressing Modes:

```
setspeed      value     R1
settime       value     R1
setweight     value     R1
setgrade      value     R1
```

- Register Addressing Modes:

```
caloxygen     Ro        R1
caldist       Ro        R1
```

Flow of Control Handling

Sequential flow of control and Branching are checked in which LABEL provides the name for a particular address where jump statements are used to change or control the flow of program.

THANK YOU!!!!