

**PRAIRIE VIEW A&M UNIVERSITY
ROY G. PERRY COLLEGE OF ENGINEERING
DEPARTMENT OF MECHANICAL ENGINEERING
ELEG 3073- Microprocessor Systems Design
FALL 2020**

I. COURSE DESCRIPTION IN ABET FORMAT

II ASSESSMENT RESULTS FOR CURRENT AND PAST THREE PERIODS

Program Outcomes Assessed in Course

Assessment Summary of Class Semester Averages for Outcomes

End of Semester Course Outcomes Assessment Reports

II. TESTS

Summary of Topics covered by Test

Original Test Questions & Master Solution/grading scheme used

(Summary of topics can be a separate page or include on the assignment itself)

Above Average (Graded original work)

Average (Graded original work)

Below Average (Graded original work)

III. HOMEWORK ASSIGNMENTS

Summary of Topics covered by Homework

Original Homework Questions & Master Solution/grading scheme used

Above Average (Graded original work)

Average (Graded original work)

Below Average (Graded original work)

IV. PROJECT

Summary of Topics Covered by Project

Project Problem Statement and Grading Scheme

Above Average (Graded original work)

Average (Graded original work)

Below Average (Graded original work)

[Include any other evaluation used and omit any method above method (Test, Quizzes, Homework, Laboratory and Project reports) that was not used]

Course Description in ABET Format

ELEG 3073 Microprocessor Systems Design

1.	Course number and name:	ELEG 3073 Microprocessor Systems Design
2.	Credits and contact hours:	3 credit hours; 3 contact hours
3.	Instructor's or course coordinator's name:	Dr. Suxia Cui
4.	Text book: Title:	Designing Embedded Systems with PIC Microcontrollers, Principles and Applications
	Author:	Tim Wilmshurst
	Publisher, Year & Edition:	Elsevier, 2 nd Edition
	a. Other Supplemental material	Instructor's handout
5.	Specific course information	
	a. Brief description of the content of the course (catalog description)	Introduction to architecture, operation, and application of microprocessors; microprocessor programming; address decoding; system timing; parallel, serial, and analog I/O; interrupts and direct memory access; interfacing to static and dynamic RAM; microcontrollers. Introduction to Microcomputers.
	b. Prerequisites or Co-requisites	ELEG 1043 and ELEG 3063. Co-requisite: ELEG 3071.
	c. *Indicate whether a required, elective, or selected elective (as per definition provided) course in the program	Required
6.	Specific goals for the course	Students will be able to:
	a. Specific outcomes of instruction (e.g. The student will be able to explain the significance of current research about a particular topic.)	design, program, and test advanced microprocessor-based systems for general purpose and/or special-purpose applications.
	b. Explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.	This course addresses: Outcome 1 Ability to identify, formulate, and solve complex engineering problems (CEP) by applying principles of engineering, science, and mathematics.
7.	Brief list of topics to be covered	Microcontroller Technology Introduction (4) PIC Instructions in Assembly Language Programming (8) Register Instruction (8) I/O and Subroutines and Time Delay Routines (8) PIC Microcontroller Simple I/O and WDT (Watch Dog Timer) Controls Exercises (4) I/O Controls with Interrupts Applications (4) Controls and LCD Module Parallel Communications and Table Lookup

		(4) Matrix Keypad Software Scanning: Debouncing & Decoding Design (4) Introduction to HPC	(4)

II ASSESSMENT RESULTS FOR CURRENT AND PAST THREE PERIODS

Program Outcomes Assessed in this course

Outcome 1:

An ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics.

Assessment Summary of Class Semester Averages for Outcomes (Should be provided for Fall 2019 through Spring 2022 Assessment Period)

Outcomes	1
Semester	Semester Class Average (%)
Fall 2020	ELEG: 71.3; CPEG: 88
Spring 2020	ELEG: 82; CPEG: 83
Fall 2019	ELEG: 79; CPEG: 77
Comments on Trend of Data (Improving or deteriorating)	ELEG students' performance improved then dropped. CPEG Students' performance improved.

*Expected Semester Class Average = 70% (or as defined by the program)
 Ideal Semester Class Average = 75% (or as defined by the program)

End of Semester Course Assessment Reports Fall 2020 and Last Three Assessment Periods.

Each summary contains the following Information

1. Performance Statistics

- Total number of students in class
- The class average performance (**stated as %**) in each course outcome assessed in this class and the **two previous times course was assessed**.
- The acceptable class average for the class in each outcome area as set by your Department.
- Percentage of students who scored below the expected average in each outcome.

2. Implementation Summary

- Brief summary of plans implemented during the semester based on last assessment report

3. Perceived Problems

- Instructor's **critical evaluation** of perceived problems that affected students' performance. **Realistic problems should be identified throughout the semester and summarized in this section of the report.**

4. New Plans for Addressing Problems

- Instructor's plans for addressing the perceived problems the next time the class is taught. **These should be specific enough for another instructor to implement. Most of these should be things that, you, the instructor can implement to help students to improve. Be creative.**

5. Overall Trend over last three assessment periods

- Instructors summary of overall performance trend for each outcome measured over the last three periods.

5. Were Expectations Met?

- Answer **YES** if the **class average was above the expected average** and the **stated percentage of students** scored at or above the expected average, else answer NO.

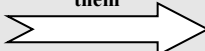
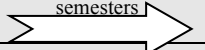
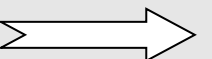
END OF SEMESTER COURSE OUTCOME ASSESSMENT REPORT
ELEG 3073-Z01 MICROPROCESSOR SYSTEMS DESIGN (ELEG)
FALL 2020 SEMESTER

Report Prepared by: Dr. Suxia Cui

Report Date: Dec. 18, 2020

Semester	Analysis Type	Number of Students	Sub-Outcome 1.1 Identify and Formulate engineering/technical/computing problems using principles of Engineering, mathematics, science				Sub-Outcome 1.2 Solve CEP/computing problems				Outcome 1 An ability to identify, formulate, and solve complex engineering problems (CEP) by applying principles of engineering, science, and mathematics.			
			Expected Class Average /	Expected Percent of students at or above average	Class Average	% Students meeting Expected Average	Expected Class Average /	Expected Percent of students at or above average	Class Average	% Students meeting Expected Average	Expected Class Average /	Expected Percent of students at or above average	Class Average	% Students meeting Expected Average
Fall 2020	Direct	7	75	70	77	86	75	70	62	43	75	70	71.3	71
Spring 2020	Direct	24	75	70	78	90	75	70	79	90	75	70	82	100
Spring 2019	Direct	21	75	70	N/A	100	75	70	N/A	N/A	75	70	79	100
Implementation Summary	Items implemented from previous report or after meeting previous instructor		Students were given guidance and sample topics from previous years' students' work. They were able to identify real world problems for an engineering solution.				The use of microcontroller skills in design analysis was emphasized. Students were able to complete the design, and successfully solve the problem.				Most students fulfilled the requirements of this outcome.			
Perceived Problems	Perceived problems are directly related to the PC and the sub items under them		One student didn't turn in the assignment.				One student didn't attach the assignment. Remote learning gave challenges to hands-on experiments.				Some students didn't finish the project.			
Plans for Addressing Problems	Plans for address specific measures to address PC and their sub areas		Introduce more methods to illustrate the engineering solutions.				Emphasize hands-on experiments during semester.				All of the plans suggested for the sub-outcomes will address the problems encountered for this outcome.			
Overall Trend over Periods	Ascertain if there were improvements over previous semesters		Students' performance dropped a little.				Students' performance dropped from last semester.				Students performance dropped.			
Were Expectations Met?			YES				NO				YES			

END OF SEMESTER COURSE OUTCOME ASSESSMENT REPORT
ELEG 3073-Z01 MICROPROCESSOR SYSTEMS DESIGN (CPEG)
FALL 2020 SEMESTER
Report Prepared by: Dr. Suxia Cui
Report Date: Dec. 18, 2020

Semester	Analysis Type	Number of Students	Sub-Outcome 1.1 Identify and Formulate engineering/technical/computing problems using principles of Engineering, mathematics, science				Sub-Outcome 1.2 Solve CEP/computing problems				Outcome 1 An ability to identify, formulate, and solve complex engineering problems (CEP) by applying principles of engineering, science, and mathematics.			
			Expected Class Average /	Expected Percent of students at or above average	Class Average	% Students meeting Expected Average	Expected Class Average /	Expected Percent of students at or above average	Class Average	% Students meeting Expected Average	Expected Class Average /	Expected Percent of students at or above average	Class Average	% Students meeting Expected Average
Fall 2020	Direct	6	75	70	90	100	75	70	84	87.5	75	70	88	100
Spring 2020	Direct	13	75	70	79	100	75	70	79.8	92	75	70	83	92
Spring 2019	Direct	5	75	70	N/A	100	75	70	N/A	N/A	75	70	77	100
Implementation Summary	Items implemented from previous report or after meeting previous instructor		Students were given guidance and sample topics from previous years' students' work. They were able to identify real world problems for an engineering solution.				The use of microcontroller skills in design analysis was emphasized. Students were able to complete the design, and successfully solve the problem.				Most students fulfilled the requirements of this outcome.			
Perceived Problems	Perceived problems are directly related to the PC and the sub items under them 		Some students' reports need improvement.				Online teaching and laboratory gave challenges to students to perform hands-on projects.				Some students didn't include enough details in the reports to explain the sub-outcomes.			
Plans for Addressing Problems	Plans for address specific measures to address PC and their sub areas 		Introduce more methods to illustrate the engineering solutions.				Emphasize hands-on experiments during semester.				All of the plans suggested for the sub-outcomes will address the problems encountered for this outcome.			
Overall Trend over Periods	Ascertain if there were improvements over previous semesters 		Students' performance is improving.				Students' performance is improving.				Students performance improved.			
Were Expectations Met?			YES				YES				YES			

MICROPROCESSOR SYSTEMS DESIGN FALL 2020

Title of Assignment: MIDTERM EXAM

TOPICS COVERED IN MIDTERM EXAM

Midterm exam covered the following topics:

1. Concepts of Microprocessor and Microcontroller
2. Architecture
3. Different number systems
4. Assembly language

ELEG3073 Microprocessor Systems Design
Midterm Exam Solution

Name: Score:

1. (40 points) Answer below questions.
 - a. (5 points) Use 8 bits to represent the 2's complement of -8. Show detail steps.

1111 1000
 - b. (5 points) Convert the following from binary to decimal.
 $101.101_2 = 5.625_{10}$
 - c. (5 points) Convert the following from hexadecimal to BCD.
 $3A_{16} = 58_{10} = 0101\ 1000_{BCD}$
 - d. (5 points) Find the binary, octal, and hexadecimal of 18_{10} . (Show procedures)
 $18_{10} = 10010_2 = 22_8 = 12_{16}$
 - e. (5 points) State four fields in the Assembly language format.
Label, Operation (OpCode), Operand, Comment
 - f. (5 points) How many bytes are there in PIC16F84A programming memory, data memory? And how many bytes are for SFR and GPR?

(Learning module 1 page 18; textbook page 37 & 38)

PRAM (programming flash memory) of **1024 (1K)** words: Range from 0X0000 - 0X03FF. Used in PIC16X84A

Data Memory: There are **84 bytes of DM** (data memory) in categories of:

16 bytes Special Function Registers (SFR)

68 bytes General Purpose Registers (GPR)

- g. (5 points) What is CISC? What is RISC? (Give the full names and also compare them.)

Complex Instruction Set Computer (CISC)

This instruction set has many instructions with a very high degree of sophistication. Each instruction usually takes several clock cycles to execute. In fact, the very sophistication which the instruction type is designed for seriously slows the operation of user systems down. Some of the instructions can be characterized with a single byte of data, while others take several bytes to define them, thus taking longer to execute. It can take as many as 8 clock cycles to execute a single instruction.

Reduced Instruction Set Computer (RISC)

This instruction set consists of fewer, simpler, less sophisticated instructions. This leads to much faster execution than **CISC** systems. One important feature of the **RISC** system is that each instruction can be characterized by a single byte of data. The other major feature is that most **RISC** instructions execute in a single clock cycle.

h. (5 points) List three different addressing modes.

There are three addressing modes used by PIC processors:

- literal (immediate),
- direct addressing, and
- indirect addressing.

(Learning module 2-2 page 32)

2. (10 points) Explain what will happen after executing the following assembly codes?

- a. BTFSC STATUS, 3
GOTO LOOP
GOTO MAIN
(before executing the STATUS = 0X07, PC = Address)

Based on the bit 3 of STATUS register, if it is 0, program will skip next instruction and perform GOTO MAIN, if it is 1, program will perform GOTO LOOP.

Because 0X08 = 0000 0111₂ bit 3 is 0, so PC = MAIN address
(No flags will be affected.)

- b. DECF COUNTER, F
(before executing the COUNTER = 0X01)

Decrease the value stored in COUNTER by 1. Result stores in COUNTER register.
(Z flag will be affected.)

Because COUNTER = 0X01, after execution, COUNTER = 0X00, Z = 1

3. (10 points) Briefly explain the following assembly codes. Specify what value (in hexadecimal) will be stored in W register after execution? Also indicate the C & Z & DC flags status after each operation if influenced.

- a. ANDWF TEMP, W
(before executing W = 0X2A, TEMP = 0XC4)

Bit by bit AND values in TEMP with W, result in W. Only Z bit is influenced.

W = 0X2A = 0010 1010₂

TEMP = 0XC4 = 1100 0100₂

Result = 0000 0000₂ = 0X00

After: W = 0X00, (TEMP = 0XC4), Z = 1

- b. SUBWF TEMP, W
(before executing W = 0X1B, TEMP = 0X36)

Subtract W from TEMP register, result stores in W. All Z, C, and DC flags will be affected.

Since $TEMP = 0X36 = 0011\ 0110_2$

$W = 0X1B = 0001\ 1011_2 \rightarrow 1110\ 0100_2 + 1 \rightarrow 1110\ 0101_2$ (2's complement of $-0X1B$)

$0011\ 0110$

$+ 1110\ 0101$

$10001\ 1011 \rightarrow 0X1B$

So $W = 0X1B$, ($TEMP = 0X36$), $Z = 0$, $C = 1$, $DC = 0$

4. (10 points) Find the total time delay of the following program when it is done. (Note: This microcontroller is running on 20MHz resonator.)

```
    MOVLW0X10
MOVWF COUNT
NOP
    LI DECF  COUNT, F
    NOP
BTFSS  STATUS, Z
GOTO   L1
```

```
    MOVLW0X10    (1)
MOVWF COUNT    (1)
NOP            (1)
    LI DECF  COUNT, F  (1)
    NOP            (1)
BTFSS  STATUS, Z  (1 or 2)
GOTO   L1    (2)
```

$20\text{MHz} / 4 = 5\ \text{MHz}$

Instruction cycle = $1/5\text{MHz} = 0.2\ \mu\text{s}$

Total cycles = $1+1+1+[16(1+1+1+2)-1] = 3+(16 \times 5-1) = 3+79 = 82$

Total delay time = $0.2 \times 82 = 16.4\ \mu\text{s}$

5. (10 points) Write the Assembly code that can:
- Initialize **PORTB: RB7, RB5, RB3, and RB1** as **input** pins and **RB6, RB4, RB2, and RB0** as **output** pins in the PIC16F84A?
 - Change **RB5** to output and **RB0** to input, others remain the same.
(Don't forget bank shift)

REGISTER FILE MAP - PIC16F84A

Address		Address
00h	Indirect addr. ⁽¹⁾	80h
01h	TMR0	81h
02h	PCL	82h
03h	STATUS	83h
04h	FSR	84h
05h	PORTA	TRISA
06h	PORTB	TRISB

STATUS Register (Read/Write) (0X03, 0X83)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	IRP	RP1	RP0	TO	PD	Z	DC	C
RESET	0	0	0	1	1	X	X	X

Starting at the bottom of the Status Register descriptions:

RP1 RP0 Register Bank Select Bits (used for direct addressing)

For PIC16X84:

0	0	Bank 0 (0X00 - 0X7F) Only 0X00-0X4F implemented by PIC16X84
0	1	Bank 1 (0X80 - 0XFF) Only 0X80-0XCF implemented by PIC16X84

a. BCF STATUS, RP0
 CLRF PORTB
 BSF STATUS, RP0
 MOVLW 0XAA
 MOVWF TRISB
 BCF STATUS, RP0

b. BSF STATUS, RP0
BCF TRISB, 5
BSF TRISB, 0
 BCF STATUS, RP0

(Or similar to part a, but MOVLW 0X8B)

6. (10 points) Below assembly codes generate a series numbers in DM address. Please fill out the empty table to illustrate the numbers in DM (addresses match the values).

```

MOVLW0X18
MOVWFFSR
MOVLW0X26
MOVWFINDF
MOVWFTEMP
LOOP
MOVF TEMP, W
INCF FSR, F
  
```



```

ADDLW 0X03
MOVWF TEMP
MOVWF INDF
MOVLW 0X21
SUBWF FSR, W
BTFSS STATUS, Z
GOTO LOOP

```

Address	Value
0X18	0X26
0X19	0X29
0X1A	0X2C
0X1B	0X2F
0X1C	0X32
0X1D	0X35
0X1E	0X38
0X1F	0X3B
0X20	0X3E
0X21	0X41

7. (10 points) Translate assembly language into machine language: ADDWF PCL, F

00 0111 1000 0010

Because: operation code = 00 0111

d = F = 1

PCL = 0x02 = 000 0010

ABOVE AVERAGE

Score for this quiz: 70 out of 100

Submitted Oct 7 at 9:30am

This attempt took 90 minutes.

Question 1

/ 5 pts

Use 8 bits to represent the 2's complement of -8. Show detail steps.

Your Answer:

$$1000_2 = 8_{10}$$

$$11000_2 = -8_{10}$$

$$00011000 = -8 \text{ (8bit)}$$

$$11111000 = -8 \text{ in 2's compliment}$$

Additional Comments:

Question 2

/ 5 pts

Convert the following from binary to decimal

$$101.101_2 =$$

Your Answer:

$$5.625_{10}$$

Additional Comments:

Question 3

/ 5 pts

Convert the following from hexadecimal to BCD.

$$3A_{16}$$

Your Answer:

$$3A_{16}$$

$$3A_{16} \text{ in binary: } 00111010_2$$

Which converts to 58_{10}

$$\text{BCD} = 01011000$$

Additional Comments:

Question 4

/ 5 pts

Find the binary, octal, and hexadecimal of 18_{10} . (Show procedures)

$$18_{10} =$$

Your Answer:

18₁₀

In binary: 10010₂

In octal: 22₈

In Hexadecimal: 12₁₆

Additional Comments:

Question 5

/ 5 pts

State four fields in the Assembly language format.

Your Answer:

Label, Operand, Comment, Mnemonic

Additional Comments:

Question 6

/ 5 pts

How many bytes are there in PIC16F84A programming memory, data memory? And how many bytes are for SFR and GPR?

Your Answer:

PIC16F84A: 1,000 Program Memory, 68 Bytes Data Memory

SFR: Default = 0 , another = up to 128

GPR: Default = 0 , another = up to 128

Additional Comments:

Question 7

/ 5 pts

What is CISC? What is RISC? (Give the full names and also compare them.)

Your Answer:

CISC: Complex Instruction Set Computer

RISC: Reduced Instruction Set Computer

CISC is most often used in automation devices and is slower than RISC chips. Also, CISC has an emphasis on Hardware. RISC has an emphasis on Software and is used in video and image processing applications.

Additional Comments:

Question 8

/ 5 pts

List three different addressing modes.

Your Answer:

Direct Addressing, Indirect Addressing, Indexed Addressing

Additional Comments:

Question 9

/ 5 pts

Explain what will happen after executing the following assembly codes?

BTFSC STATUS, 3

GOTO LOOP

GOTO MAIN

(before executing the STATUS = 0X07, PC = Address)

Your Answer:

Operation: Skip if (f) = 0. If bit b in register F is '0' then next instruction is skipped. If b is 1 execute next instruction

Status = 0X07 = 00000111

Here bit 3 = 0

SKIP.

Additional Comments:

Question 10

/ 5 pts

Explain what will happen after executing the following assembly codes?

DECF COUNTER, F

(before executing the COUNTER = 0X01)

Your Answer:

Counter - 1 = 0X01 - 1 = 0X0

After:

Counter = 0X00

Additional Comments: Z=?

Question 11

/ 5 pts

Briefly explain the following assembly codes. Specify what value (in hexadecimal) will be stored in W register after execution? Also indicate the C& Z & DC flags status after each operation if influenced.

ANDWF TEMP, W

(before executing W = 0X2A, TEMP = 0XC4)

Your Answer:

Temp = 0XC4

W = 0X00

Additional Comments: Z=?

Question 12

/ 5 pts

Briefly explain the following assembly codes. Specify what value (in hexadecimal) will be stored in W register after execution? Also indicate the C& Z & DC flags status after each operation if influenced.

SUBWF TEMP, W

(before executing W = 0X1B, TEMP = 0X36)

Your Answer:

W = 0X1B

TEMP = 0X36

Additional Comments: flag bits?

Question 13

/ 10 pts

Find the total time delay of the following program when it is done. (Note: This microcontroller is running on 20MHz resonator.)

```
        MOVLW      0X10
        MOVWF      COUNT
        NOP
L1:     DECF        COUNT, F
        NOP
        BTFSS      STATUS, Z
        GOTO       L1
```

Your Answer:

$$20\text{MHz}/4 = 5/\text{MHz}$$

$$L1 = (1+1+1+(16(1+1+1+2)-1))$$

$$= 3 + (16 \cdot 5) - 1$$

$$= 3 + 80 - 1 = 82 \text{ bytes}$$

$$\text{Total Delay Time: } 82 \text{ bytes}/5\text{MHz} = 1.64 \times 10^{-5} \text{ Sec}$$

Additional Comments:

Question 14

/ 10 pts

Write the Assembly code that can:

1.

1. Initialize **PORTB: RB7, RB5, RB3, and RB1** as **input** pins and **RB6, RB4, RB2, and RB0** as **output** pins in the PIC16F84A?

2. Change **RB5** to output and **RB0** to input, others remain the same.

(Don't forget bank shift)

Your Answer:

1.

RB7:1

RB6:0

RB5:1

RB4:0

RB3:1

RB2:0

RB1:1

RB0:0

BFC STATUS RPO

CLRF PORT B

BSF STATUS RPU

MOVLW 0XAA

2.

RB7:1

RB6:0

RB5:0

RB4:0

RB3:1

RB2:0

RB1:1

RB0:1

BCF STATUS RPO

CLRF PORT B

BSG STATUS RPU

MOVLW 0XBB

MOVWF TRISB

MOVWF TRIS B

BCF STATUS RPU

BCF STATUS RPO

Additional Comments:

UnansweredQuestion 15

/ 10 pts

Below assembly codes generate a series numbers in DM address. Please fill out the empty table to illustrate the numbers in DM (addresses match the values).

MOVLW	0X18
MOVWF	FSR
MOVLW	0X26
MOVWF	INDF
MOVWF	TEMP

LOOP

MOVF	TEMP, W
INCF	FSR, F
ADDLW	0X03
MOVWF	TEMP
MOVWF	INDF
MOVLW	0X21
SUBWF	FSR, W
BTFSS	STATUS, Z
GOTO	LOOP

Address

Value

Your Answer:
Additional Comments:

Unanswered Question 16
/ 10 pts

Translate assembly language into machine language: ADDWF PCL, F
Your Answer:

AVERAGE

Score for this quiz: 53 out of 100

Submitted Oct 7 at 9:27am

This attempt took 87 minutes.

Question 1

/ 5 pts

Use 8 bits to represent the 2's complement of -8. Show detail steps.

Your Answer:

11111000

Additional Comments:

Question 2

/ 5 pts

Convert the following from binary to decimal

$101.101_2 =$

Your Answer:

5.625

Additional Comments:

Question 3

/ 5 pts

Convert the following from hexadecimal to BCD.

$3A_{16}$

Your Answer:

01011000

Additional Comments:

Question 4

/ 5 pts

Find the binary, octal, and hexadecimal of 18_{10} . (Show procedures)

$18_{10} =$

Your Answer:

binary= 10010(base 2)

octal=22(base 8)

hex=12(base 16)

Additional Comments:

Question 5

/ 5 pts

State four fields in the Assembly language format.

Your Answer:

label, operation, operand, and comment

Additional Comments:

Question 6

/ 5 pts

How many bytes are there in PIC16F84A programming memory, data memory? And how many bytes are for SFR and GPR?

Your Answer:

68 bytes in PIC16F84A

SFR default bank= 0

Another SFR bank= up to 128 bytes

GPR default bank= 0

Another GPR bank= up to 128 bytes

Additional Comments:

Question 7

/ 5 pts

What is CISC? What is RISC? (Give the full names and also compare them.)

Your Answer:

CISC- Complex Instruction Set Computing

RISC- Reduced Instruction Set Computing

RISC is more focused on software and used in image/video processing applications, where CISC has complete emphasis on Hardware and consists of single clock cycles which execute only one instruction.

Additional Comments:

Question 8

/ 5 pts

List three different addressing modes.

Your Answer:

Direct, Indirect, and Implicit addressing mode

Additional Comments:

Question 9

/ 5 pts

Explain what will happen after executing the following assembly codes?

BTFSC STATUS, 3

GOTO LOOP

GOTO MAIN

(before executing the STATUS = 0X07, PC = Address)

Your Answer:

BTFSC- Bit Test File, Skip if Color

the operation will skip (t(b))=0, and if bit b in register f is 0 then the next instruction would be skipped. On the other hand if b is 1 then the next instruction will be executed.

Additional Comments:

Question 10

/ 5 pts

Explain what will happen after executing the following assembly codes?

DECF COUNTER, F

(before executing the COUNTER = 0X01)

Your Answer:

The operation will be f - 1- xf where the counter = 0x01 and the counter-1= 0x01-1 which is = 0x0.

After executing the instruction the counter= 0x0 and result is 0, and zero flag is set to z = 1

Additional Comments:

Question 11

/ 5 pts

Briefly explain the following assembly codes. Specify what value (in hexadecimal) will be stored in W register after execution? Also indicate the C& Z & DC flags status after each operation if influenced.

ANDWF TEMP, W

(before executing W = 0X2A, TEMP = 0XC4)

Your Answer:

These assembly code instructions performs an AND logical operation between the "W" registered value and the "f" file registered. If you get d is 0 as a result, the function result value will be stored in the W register and if d is 1 then the result will stored into the file register instead. The status flags affected are Z(zero flag) where Z=0 if result does not equal 0 and Z=1 if the result is in fact 0. we are given W= 0X2A and TEMP = 0XC4, therefore ANDing of both will be d = W = 0X2A AND 0XC4

100010

AND

110010110

Additional Comments:

Question 12

/ 5 pts

Briefly explain the following assembly codes. Specify what value (in hexadecimal) will be stored in W register after execution? Also indicate the C& Z & DC flags status after each operation if influenced.

SUBWF TEMP, W

(before executing W = 0X1B, TEMP = 0X36)

Your Answer:

The following assembly codes sets instruction to subtract the contents of the W register value from the contents of "f" file register. If d is 0, the result value will be stored in the W register and if d is 1 the result is stored into the "f" file register in return. The operation will thus be: $f - W > d$.

The status flags affected: Z(zero flag), C(carry flag) and DC(Digit Carry flag) therefore,

Z=0 if the subtraction result is not 0

Z=1 if subtraction result is 0

C=0 if the borrow occurs

C=1 if borrow did not occur

DC=0 if borrow from the low order 4 bits did in fact occur

DC=1 if borrow from the low order 4 bits did not occur

$11011 - 110110 = -011011$ (binary form)

Additional Comments: So calculated result =?

Question 13

/ 10 pts

Find the total time delay of the following program when it is done. (Note: This microcontroller is running on 20MHz resonator.)

MOVLW 0X10

MOVWF COUNT

NOP

L1	DECF	COUNT, F
	NOP	
	BTFSS	STATUS, Z
	GOTO	L1

Your Answer:

MOVLW 0X10 > $1 \times (5 \times 10^{-7}) = 5 \times 10^{-7}$

MOVWF > $1 \times (5 \times 10^{-7}) = 5 \times 10^{-7}$

NOP > $1 \times (5 \times 10^{-7}) = 5 \times 10^{-7}$

DECF > $18 \times (5 \times 10^{-7}) = 9 \times 10^{-6}$

NOP > $18 \times (5 \times 10^{-7}) = 9 \times 10^{-6}$

BTFSS > $36 \times (5 \times 10^{-7}) = 1.8 \times 10^{-5}$

GOTO > $36 \times (5 \times 10^{-7}) = 1.8 \times 10^{-5}$

total delay = 5.55×10^{-5}

Additional Comments:

UnansweredQuestion 14

/ 10 pts

Write the Assembly code that can:

1.

1. Initialize **PORTB: RB7, RB5, RB3, and RB1** as **input** pins and **RB6, RB4, RB2, and RB0** as **output** pins in the PIC16F84A?

2. Change **RB5** to output and **RB0** to input, others remain the same.

(Don't forget bank shift)

Your Answer:

Additional Comments:

UnansweredQuestion 15

/ 10 pts

Below assembly codes generate a series numbers in DM address. Please fill out the empty table to illustrate the numbers in DM (addresses match the values).

MOVLW	0X18
-------	------

	MOVWF	FSR
	MOVLW	0X26
	MOVWF	INDF
	MOVWF	TEMP
LOOP		
	MOVF	TEMP, W
	INCF	FSR, F
	ADDLW	0X03
	MOVWF	TEMP
	MOVWF	INDF
	MOVLW	0X21
	SUBWF	FSR, W
	BTFSS	STATUS, Z
	GOTO	LOOP

Address

Value

Your Answer:

Additional Comments:

Unanswered Question 16

/ 10 pts

Translate assembly language into machine language: ADDWF PCL, F

Your Answer:

Additional Comments:

BELOW AVERAGE

Score for this quiz: 42 out of 100

Submitted Oct 7 at 9:24am

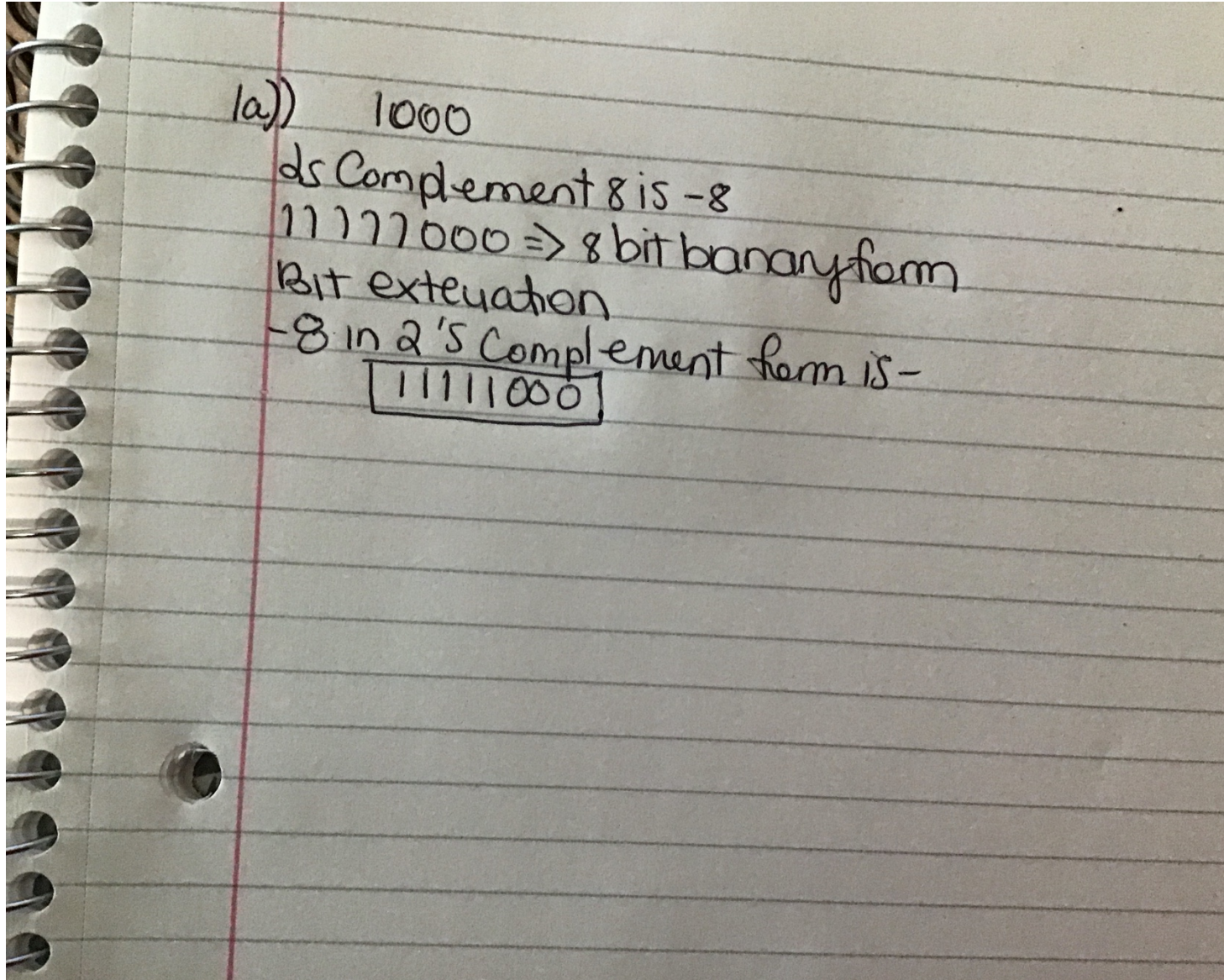
This attempt took 81 minutes.

Question 1

/ 5 pts

Use 8 bits to represent the 2's complement of -8. Show detail steps.

Your Answer:



Additional Comments:

Question 2

/ 5 pts

Convert the following from binary to decimal

$101.101_2 =$

Your Answer:

5.625

Additional Comments:

Question 3

/ 5 pts

Convert the following from hexadecimal to BCD.

$3A_{16}$

Your Answer:

111010

Additional Comments:

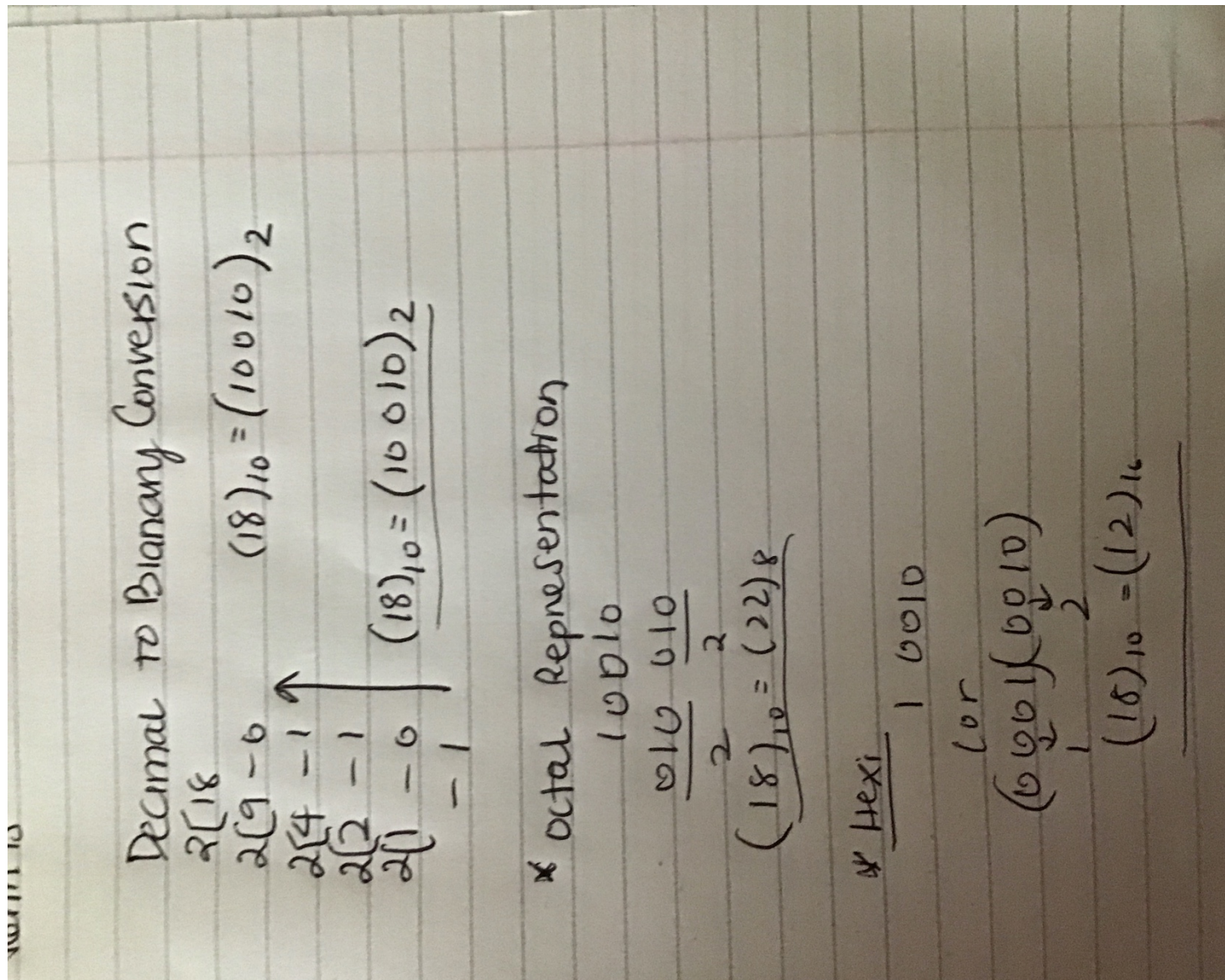
Question 4

/ 5 pts

Find the binary, octal, and hexadecimal of 18_{10} . (Show procedures)

$18_{10} =$

Your Answer:



Additional Comments:

Question 5

/ 5 pts

State four fields in the Assembly language format.

Your Answer:

Label field , operational field Opreand field And ,comment field

Additional Comments:

Question 6

/ 5 pts

How many bytes are there in PIC16F84A programming memory, data memory? And how many bytes are for SFR and GPR?

Your Answer:

⑥

PIC16F84A

X Programming memory - 1K Data memory 68 bytes SFR Bank-0
X SFR another bank, up to 128 bytes GPR - 0 GPR another Bank up to 128 bytes

Additional Comments:

Question 7

/ 5 pts

What is CISC? What is RISC? (Give the full names and also compare them.)

Your Answer:

CISC is a complex instruction set can take a while in order to configure rate there are levels to it

RISC is a less complicated system if you simply less sophisticated instructions faster execution execute in a good one clock cycle

Additional Comments:

Question 8

/ 5 pts

List three different addressing modes.

Your Answer:

are three different address in MOD their names are stated literally immediate/direct a dressing/in direct address in those are the three

Additional Comments:

UnansweredQuestion 9

/ 5 pts

Explain what will happen after executing the following assembly codes?

BTFSC STATUS, 3

GOTO LOOP

GOTO MAIN

(before executing the STATUS = 0X07, PC = Address)

Your Answer:

Additional Comments:

UnansweredQuestion 10

/ 5 pts

Explain what will happen after executing the following assembly codes?

DECF COUNTER, F

(before executing the COUNTER = 0X01)

Your Answer:

Additional Comments:

Question 11

/ 5 pts

Briefly explain the following assembly codes. Specify what value (in hexadecimal) will be stored in W register after execution? Also indicate the C & Z & DC flags status after each operation if influenced.

ANDWF TEMP, W

(before executing W = 0X2A, TEMP = 0XC4)

Your Answer:

for another bank, up to 128 bytes GPR J - 0 GPR and

⑪ ANBWf TEMP, W

Temp = 0xC4

1100 0100

C = 0

Z = 1

DC = 0

W = 0x2A

AND 0010 1010

0000 0000

0x00

⑫ SUBWF, W

W = 0x1B, Temp = 0x36

C = 0

Z = 0

DC = 0

Additional Comments: Which is the destination?

Question 12

/ 5 pts

Briefly explain the following assembly codes. Specify what value (in hexadecimal) will be stored in W register after execution? Also indicate the C & Z & DC flags status after each operation if influenced.

SUBWF TEMP, W

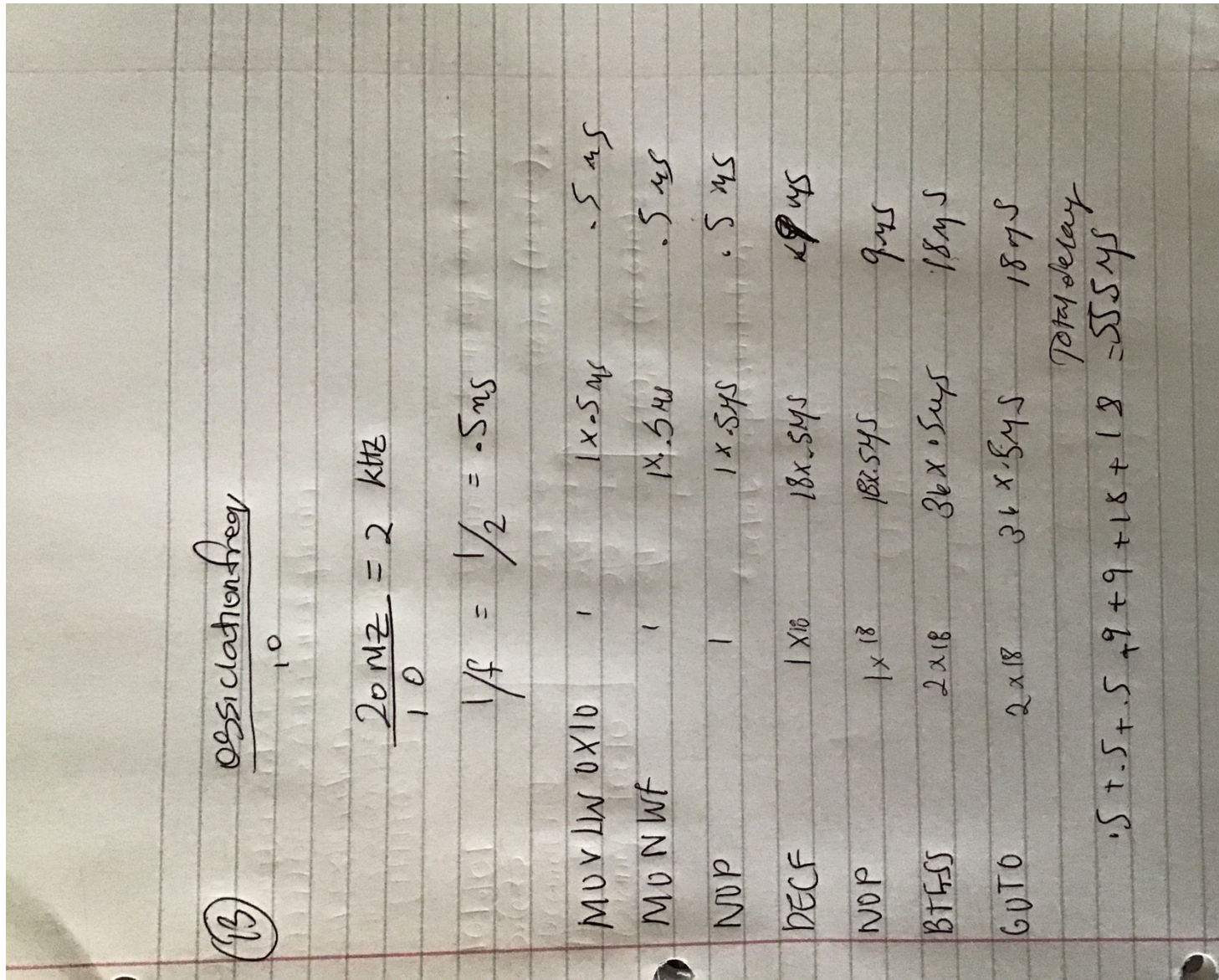
(before executing W = 0x1B, TEMP = 0x36)

Your Answer:

GOTO

L1

Your Answer:



Additional Comments:

Question 14

/ 10 pts

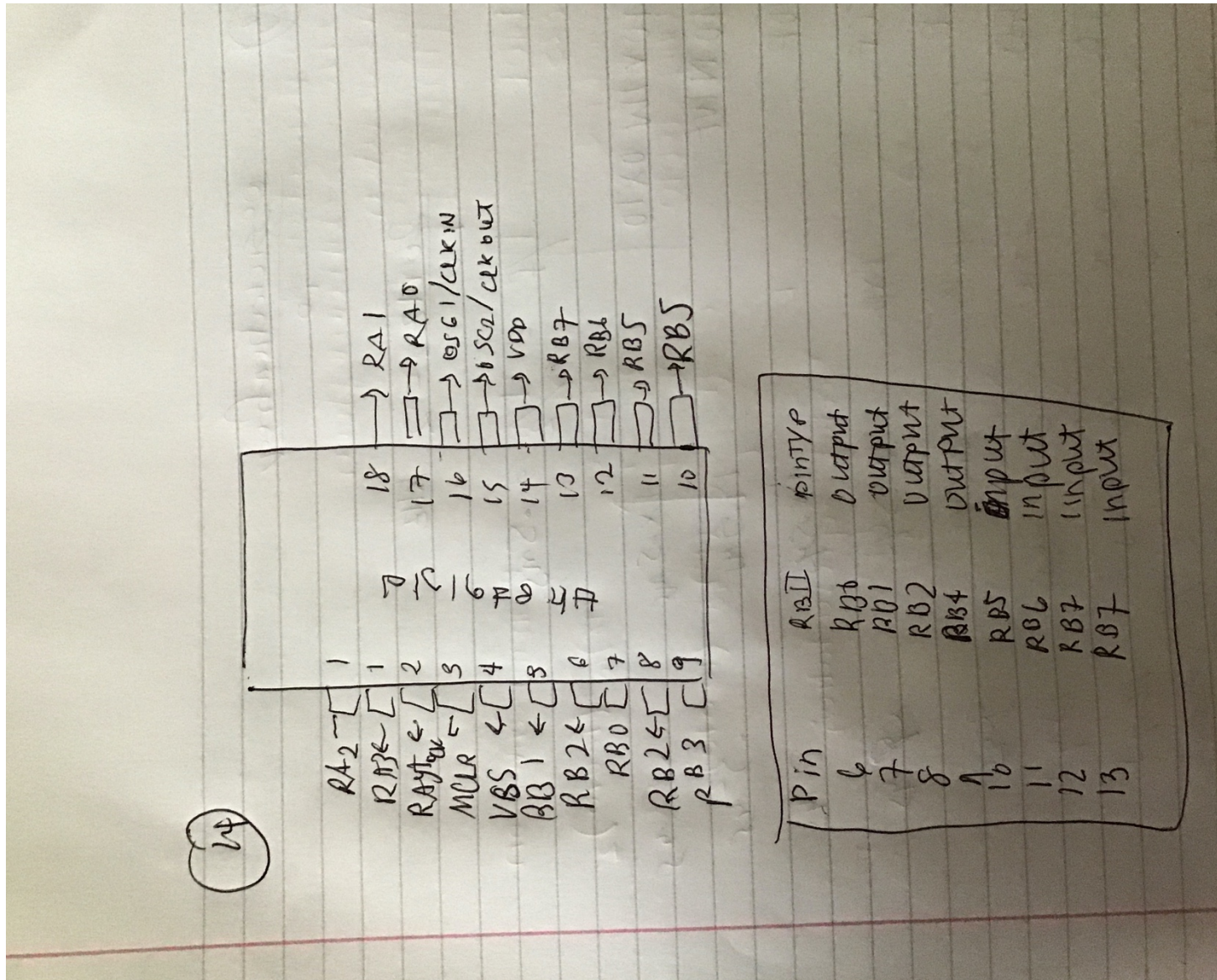
Write the Assembly code that can:

1.

1. Initialize **PORTB: RB7, RB5, RB3, and RB1** as **input** pins and **RB6, RB4, RB2, and RB0** as **output** pins in the PIC16F84A?
2. Change **RB5** to output and **RB0** to input, others remain the same.

(Don't forget bank shift)

Your Answer:



Additional Comments:

Unanswered Question 15

/ 10 pts

Below assembly codes generate a series numbers in DM address. Please fill out the empty table to illustrate the numbers in DM (addresses match the values).

MOVLW 0X18

MOVWF FSR

MOVLW 0X26

	MOVWF	INDF
	MOVWF	TEMP
LOOP		
	MOVF	TEMP, W
	INCF	FSR, F
	ADDLW	0X03
	MOVWF	TEMP
	MOVWF	INDF
	MOVLW	0X21
	SUBWF	FSR, W
	BTFSS	STATUS, Z
	GOTO	LOOP

Address

Value

Your Answer:

Additional Comments:

Unanswered Question 16

/ 10 pts

Translate assembly language into machine language: ADDWF PCL, F

Your Answer:

**MICROPROCESSOR SYSTEMS DESIGN
FALL 2020**

Title of Assignment: HOMEWORK 1

TOPICS COVERED IN HOMEWORK 1

- 1. Review logic circuits**
- 2. Review programming**

ELEG 3073 Microprocessor Systems Design

Homework 1 Solution

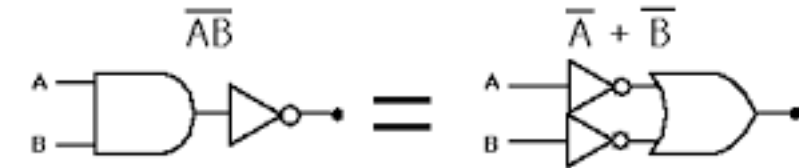
Name:

1. What is combinational logic circuit? What is sequential logic circuit?

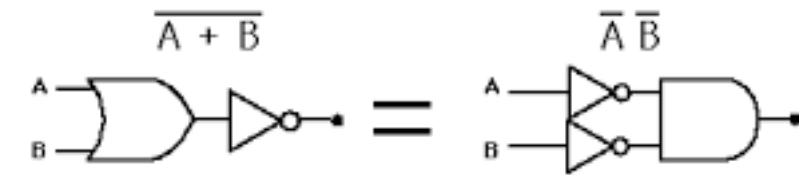
Combinational logic circuit: Output only determined by current inputs.

Sequential logic circuit: Output not only determined by current inputs, but also previous states.

2. Explain DeMorgan's Theorem?



A NAND gate is equivalent to an inversion followed by an OR



A NOR gate is equivalent to an inversion followed by an AND

3. Write the symbol and truth table of exclusive-or gate.

XOR GATE Truth Table



BOOLEAN EXPRESSION

$$A \cdot \bar{B} + \bar{A} \cdot B$$

$$(A + B) \cdot (\bar{A} + \bar{B})$$

C = A ⊕ B

Input1 Input2

Output

INPUT		OUTPUT
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

ProjectIoT123.com

4. Find the binary, octal, and hexadecimal of 28_{10} . (Show procedures)

$$28_{10} = 11100_2 = 34_8 = 1C_{16}$$

5. Find the 2's complement of -4_{10} using 8-bits. (Show procedures.)

$$4 \rightarrow 0000\ 0100 \rightarrow 1111\ 1011 + 1 \rightarrow 1111\ 1100 \text{ is the 2's complement of } -4.$$

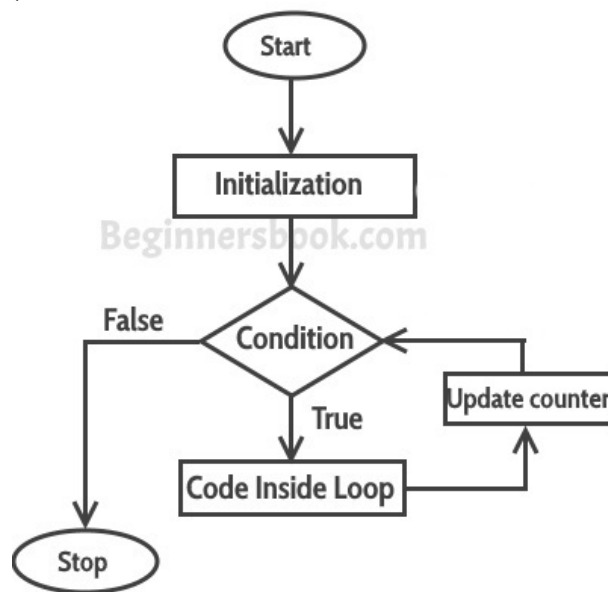
6. Write a C++ program to add integer 1 through 10. Output result to the screen. Also draw the flowchart of the program.

```
#include <iostream>
main ()
{
    int i;
    int sum = 0;
    for (i = 1; i <= 10; i++)
```

```

sum = sum+i;
printf("The sum of 1 to 10 is: %d\n", sum);
return 0;
}

```



7. Explain the differences between logic operation and bit-by-bit logic operation by given three examples.

Logical Operators

Operator	Meaning	Example	Result
&&	Logical and	$(5 < 2) \&\& (5 > 3)$	False
	Logical or	$(5 < 2) (5 > 3)$	True
!	Logical not	$!(5 < 2)$	True

Symbol ◆	Operator ◆
&	bitwise AND
	bitwise inclusive OR
^	bitwise exclusive OR
<<	left shift
>>	right shift
~	one's complement (unary)

$5 \& 2 = 0101_2 \& 0010_2 = 0000_2$
 $5 | 2 = 0101_2 | 0010_2 = 0111_2 = 7$
 $\sim 5 = \sim 0101_2 = 1010_2 = 10$

8. Did you have any experience on microprocessors/microcontrollers before? If so, which type? Please briefly introduce.

ABOVE AVERAGE

ELEG 3073 Microprocessor Systems Design

Homework 1

Name: Hyram Rounsey

6.5

1. What is combinational logic circuit? What is sequential logic circuit?

Combinational logic circuits outputs depends on the combination of its inputs. Sequential logic circuits depends on the logic states that existed previously.

2. Explain DeMorgan's Theorem?

DeMorgan's Theorem explains that the complement of the product of all the terms is equal to the sum of the complement of each term.

3. Write the symbol and truth table of exclusive-or gate.



A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

4. Find the binary, octal, and hexadecimal of 28_{10} . (Show procedures)

Binary: 11100₂ Octal: 011100₈ 34₈

Hex: 00011100₁₆ 1C₁₆

5. Find the 2's complement of -4_{10} using 8-bits. (Show procedures.)

-00000100 2's: ~11111100

6. Write a C++ program to add integer 1 through 10. Output result to the screen. Also draw the flowchart of the program.

```
int add=0;
for(int x=1; x<=10; x++)
{ add = x + add; }

cout << add << endl;
```

7. Explain the differences between logic operation and bit-by-bit logic operation by given three examples.

- Logic operation: True, False, Or, Nor operations
- Bit-by-bit operations: Not, Or, and AND Gates

8. Did you have any experience on microprocessors/microcontrollers before? If so, which type? Please briefly introduce.

~~Yes~~ Yes, I have experience with Arduino and some FPGAs

AVERAGE

Name: Omer Adnan

1. What is combinational logic circuit? What is sequential logic circuit?

Combinational circuits are defined as the time independent circuits which do not depend upon previous inputs to generate any output are termed as combinational circuits.

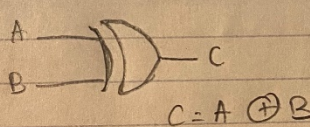
Sequential circuits are those which are dependent on clock cycles and depend on present as well as past inputs to generate any output.

2. Explain DeMorgan's Theorem?

DeMorgan's theorem states that inverting the output of any gate is the same as using the opposite type of gate with inverted inputs. In circuit terms, DeMorgan's theorem states that the AND gate with inverted output and the OR gate with inverted inputs in the following circuit are functionally equivalent.

3. Write the symbol and truth table of exclusive-or gate.

3) Exclusive OR gate



$C = A \oplus B$

input		output
A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

4. Find the binary, octal, and hexadecimal of 28_{10} . (Show procedures)

4) $28_{10} \rightarrow$ binary

Decimal	Binary	Octal	Hexa
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1101	13	B
12	1100	14	C

Handwritten conversion for 28_{10} :

$28_{10} \rightarrow$ binary

2	28	0
2	14	0
2	7	1
2	3	1
2	1	1
	0	

$\Rightarrow (11100)_2$

$(11100)_2 \rightarrow$ Octal

0	1	1	1	0	0
				3	4

$\Rightarrow (34)_8$

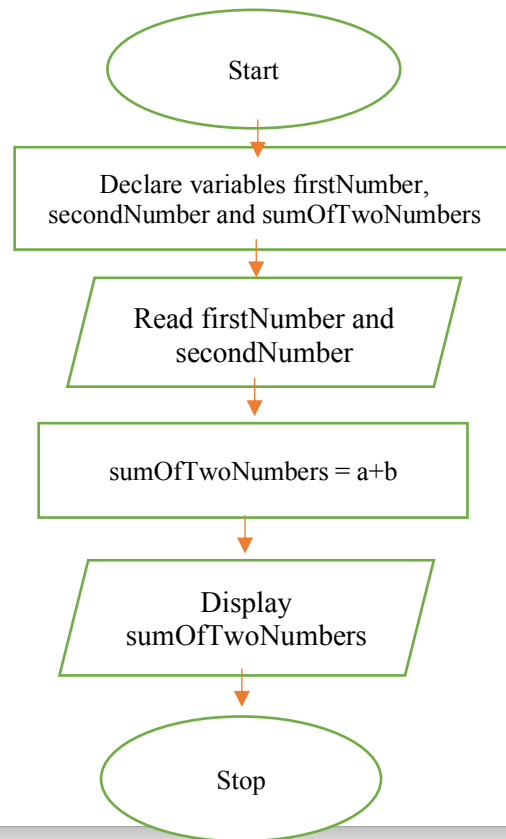
$(11100)_2 \rightarrow$ Hexadecimal

0	0	0	1	1	0	0
			1		C	

$\Rightarrow (1C)_{16}$

2^5 complement of $(-4)_{10} \rightarrow (4)_{10} \rightarrow 00000100$

6. Write a C++ program to add integer 1 through 10. Output result to the screen. Also draw the flowchart of the program.



The screenshot shows a C++ IDE with a file named 'main.cpp'. The code is as follows:

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     int firstNumber, secondNumber, sumOfTwoNumbers;
7
8     cout << "Enter two integers: ";
9     cin >> firstNumber >> secondNumber;
10
11     // sum of two numbers in stored in variable sumOfTwoNumbers
12     sumOfTwoNumbers = firstNumber + secondNumber;
13
14     // Prints sum
15     cout << firstNumber << " + " << secondNumber << " = " << sumOfTwoNumbers;
16
17     return 0;
18 }
19
```

The execution output window shows the following text:

```
"C:\Users\omera\Desktop\Labs\lab5 code\Welcome\adder\bin\Debug\ad...
Enter two integers: 3+5
3 + 5 = 8
Process returned 0 (0x0)   execution time : 3.780 s
Press any key to continue.
```

7. Explain the differences between logic operation and bit-by-bit logic operation by given three examples. A bitwise operator evaluates each bit of two expressions based on the logic that is defined by the operator. For example, for an AND gate to work, you have to define the variables A and B. Each bit of the input expressions is compared independently of other bits. A logical operator evaluates two conditions. An example of the logical operator is $\sim$ (logical not). There are two significant differences between a logical and bitwise operator. First, a logical operator will always return 1 (true) or 0 (false). Additionally, a logical operator will perform "short circuit" logic, meaning that if the outcome is known after only checking the first condition, the second condition is ignored. For example, when using the logical and, we know that both conditions must be true in order for the result to be true; if the first of the two evaluates to false, then the result will be false regardless of the second condition, and therefore the second condition will not be evaluated.

8. Did you have any experience on microprocessors/microcontrollers before? If so, which type? Please briefly introduce.

I used an Arduino Uno to build a smoke detector alarm which has built in microcontrollers.

BELOW AVERAGE

ELEG 3073 Microprocessor Systems Design

Homework 1

4.5

Name: Matthew Barnett

1. What is combinational logic circuit? What is sequential logic circuit?

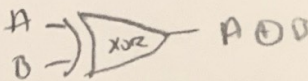
Combinational logic circuit is made with NAND, NOR, or NOT gates that are combined.

Sequential logic circuit

2. Explain DeMorgan's Theorem?

Demorgan's Theorem is mainly used to solve Boolean Algebra. it's used for implementing basic gate operation.

3. Write the symbol and truth table of exclusive-or gate.



A	B	A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

4. Find the binary, octal, and hexadecimal of 28_{10} . (Show procedures)

Binary: 11100_2

Octal: 011100_8 34_8

Hexadecimal: 0001100_{16} $1C_{16}$

5. Find the 2's complement of -4_{10} using 8-bits. (Show procedures.)

$-00000100 \rightarrow -1111100$

-4_{10}

1 0 0
4 2 1

* Flip 1's and 0's, right to left

6. Write a C++ program to add integer 1 through 10. Output result to the screen. Also draw the flowchart of the program.

```
int add = 0;
for (int x = 1; x <= 10; x++)
{
    add = x + add;
}
cout << add << endl;
```

7. Explain the differences between logic operation and bit-by-bit logic operation by given three examples.

Bit-by-bit operations
NOT, AND, OR gates

Logic operation if it is true or false.

NOR

8. Did you have any experience on microprocessors/microcontrollers before? If so, which type? Please briefly introduce.

Yes, I build computers, the CPU in the computer is a microprocessor.

**MIPROPROCESSOR SYSTEMS DESIGN
FALL 2020**

Title of Assignment: HOMEWORK 2

TOPICS COVERED IN HOMEWORK 2

- 1. Digital design**
- 2. Memory map**
- 3. Arithmetic operation**

ELEG3073 Microprocessor Systems Design

Homework 2 Solution

- Questions under "Homework" sections of Learning Module #2-1: 2, 3, 4, 5 (b) & (d)
Module 2-1

$$2. \quad 836CE_{16} = 1000\ 0011\ 0110\ 1100\ 1110_2 =$$

$$(524,288+12,288+1,536+192+14)538318_{10}=2033316_8$$

$$3. \quad 657_8 = 110101111_2 = 431_{10} = 10100110001_{BCD} = 1AF_{16}$$

$$4. \quad 110001010011001_{BCD} = 6299_{10} = 1100010011011_2 = 14233_8 = 189B_{16}$$

$$5. \quad (b) \quad 0XEF + 0XAC$$

$$1110\ 1111$$

$$1010\ 1100$$

+

$$1\ 1001\ 1011$$

$$9B$$

$$C = 1; Z = 0; DC = 1$$

$$(d) \quad 0X56 - 0XAD$$

$$0XAD = 1010\ 1101$$

$$0101\ 0010$$

$$+ \quad 1$$

$$-0XAD = 0101\ 0011$$

$$0101\ 0110$$

$$+0101\ 0011$$

$$1010\ 1001$$

$$A\ 9$$

$$C = 0; Z = 0; DC = 0$$

- Questions under "Homework" sections of Learning Module #1: 1, 2, 3

Module 1

- A microprocessor is the "brain" of typical computers and control systems; however, it can't function independently. The microprocessor must communicate or control other devices in parallel format; therefore it needs some peripheral interface chips.

- A microprocessor can't form a workable control system without the help of other peripheral devices. On the other hand, a microcontroller does have the peripheral features needed to implement the computer portion of an embedded application. Usually, the microcontroller is called a single-chip microcomputer, and it includes a microprocessor and its core, memory, and I/O facilities on the same chip.

(Learning Module page 3; Textbook page 12.)

3.

Program memory (PM, flash)

RAM File Registers (Data Memory, DM)

- Special Function Register (SFR)
- General Purpose Register (GPR)

EEPROM Data Memory

(Learning Module page 18; Textbook page 31-33)

3. Design a one bit full adder using only one type of logic gate. Draw the truth table, logic expressions, and circuit diagram.

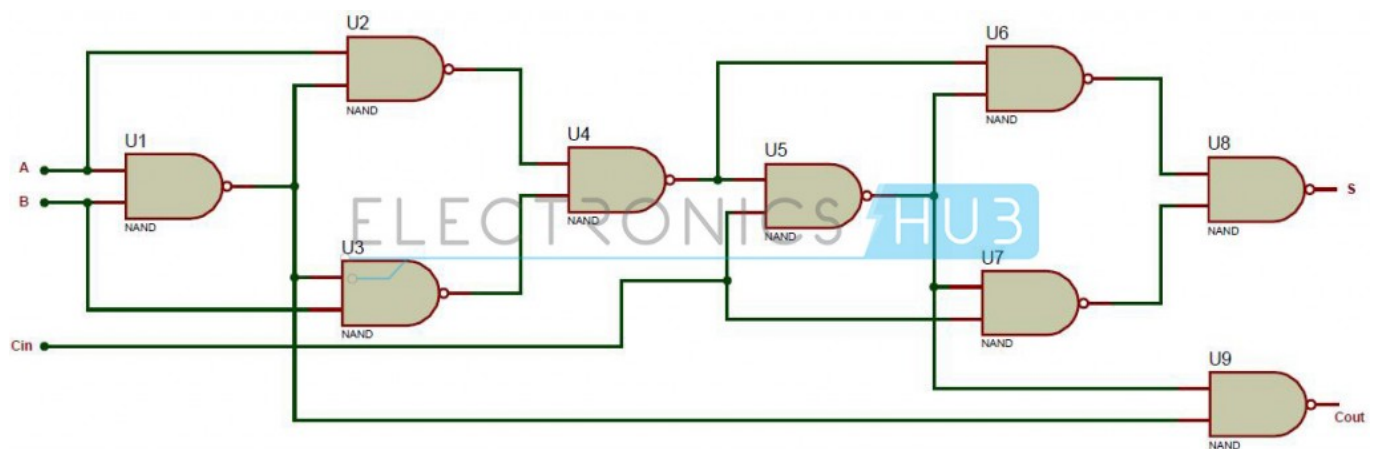
Full Adder Truth Table

The truth table for full adder is shown below.

Input			Output	
A	B	Cin	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Full Adder using NAND Gates

As mentioned earlier, a NAND gate is one of the universal gates and can be used to implement any logic design. The circuit of full adder using only NAND gates is shown below.



Full Adder using NAND Gates

4. State four fields in the Assembly language format.

(Page 6)

Label, Operation (OpCode), Operand, Comment

Label Placed in the 1st column and is the first field of a source statement. It does not have to be present, just used when it is necessary.

Operation This is a legal Operational Code mnemonic (Op code) or an assembler directive.

Operand This may contain a label, an expression, or a combination of symbols and expressions separated by commas. It is used by specifying the addressing mode of the instruction as well as the operand of the instruction.

Comment Comments are separated from the rest of the instruction by a semicolon, ";" and by at least one blank space. The compiler will ignore anything after a semicolon on that line. A ';' (semicolon) in the beginning makes the entire line a comment.

5. State the difference in DM (Data Memory) and PM (Program Memory).

Data memory is used for temporally store data or variables, it belongs to RAM, volatile.

PM is used for store program or applications, it belongs to ROM or flash, non-volatile.

6. How many bytes are there in PIC16F84A programming memory, data memory? And how many bytes are for SFR and GPR?

(Learning module 1 page 18; textbook page 37 & 38)

PRAM (programming flash memory) of **1024 (1K)** words: Range from 0X0000 - 0X03FF.

Used in PIC16X84A

Data Memory: There are **84 bytes of DM** (data memory) in categories of:

16 bytes Special Function Registers (SFR)

68 bytes General Purpose Registers (GPR)

7. How many bytes are there in PIC16F88 programming memory, data memory? And how many are for SFR and GPR?

(Learning module 1 page 18-19)

There are **4096 (4k)** words of PM (programming flash memory) Address range from 0X0000 - 0X0FFF. Used in the PIC16F88.

There are **441 bytes** of DM (data memory) in categories of:

73 bytes Special Function Register (SFR)

368 bytes General Purpose Register (GPR)

8. How many bytes are there in PIC16F877A programming memory, data memory? And how many are for SFR and GPR?

(Learning module 1 page 19)

There are **8192 (8k)** words of PM (programming flash memory) Address range from 0X0000 - 0X1FFF. Used in the PIC16F877A.

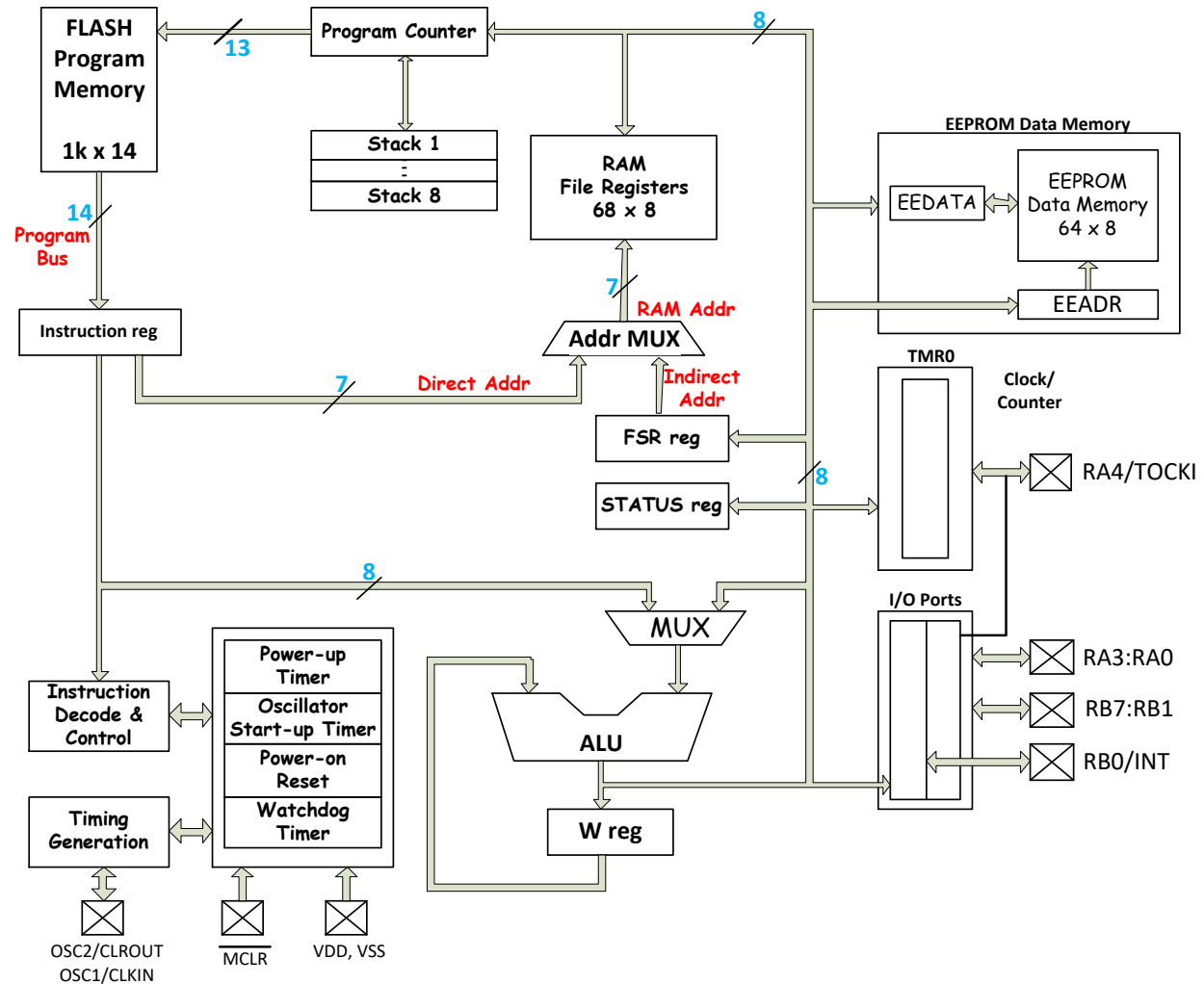
There are **464 bytes** of DM (data memory) in categories of:

96 bytes Special Function Register (SFR)

368 bytes General Purpose Register (GPR)

9. Which file (extension name) is to be download to the PIC16F MCU while using PICKIT2 software?
 .HEX file.

10. Draw the block diagram for PIC 16F84A.



ABOVE AVERAGE

Gabriel Maxwell

ELEG 3073

Homework 2

Module #2-1

1.

2. 8 3 6 C E Binary, Decimal and Octal

1000 0011 0110 1100 1110

★ 10000011011011001110₂ Binary $2^8 2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0 2^9 2^8 2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$ $512 + 256 + 128 + 64 + 32 + 16 + 8 + 4 + 2$ $(538318)_{10}$ Decimal $(203316)_8$ Octal3. 657₈ Binary, BCD, Decimal, and Hex $110101111 \Rightarrow (110101111)_2$ Binary $2^5 + 2^7 + 2^5 + 2^2 + 2^1 + 2^0 = (431)_{10}$ Decimal $431_{10} \Rightarrow (0100, 0011, 0001)_{BCD}$

1

A

F

 $431_{10} \Rightarrow 1AF_{16}$ Hex4. 110001010011001_{BCD}

0110 0010 1001 1001

6 2 9 9

 $\Rightarrow (6299)_{10}$ Decimal

0001 1000 1001 1011

1 8 9 3

 $\Rightarrow (1893)_{10}$ Hex

001 100 010 011 011

1 4 2 3 3

Octal

 $\Rightarrow (14233)_8$

(14233)

2 | 6299

2 | 3149 1

2 | 1574 1

2 | 787 0

2 | 393 1

2 | 196 1

2 | 98 0

2 | 49 0

2 | 24 1

2 | 12 0

2 | 6 0

2 | 3 0

2 | 1 1

2 | 0 1

2 | 0 1

2 | 0 1

2 | 0 1

2 | 0 1

2 | 0 1

Binary

 $\Rightarrow (110001001101)_2$

Gabriel Maxwell
ELEG 3023
Homework 2

B
5. $0xEF + 0xAC$
 $(EF_{16}) + (AC_{16})$
 16, 17 12, 14
 $1110 - 1111 + 1010 - 1100$

$$\begin{array}{r} 1110 \quad 1111 \\ + 1010 \quad 1100 \\ \hline 1001 \quad 1011 \end{array} \Rightarrow \begin{array}{l} \text{Carry Flag} = 1 \\ \text{Zero Flag} = 0 \\ \text{Auxiliary Flag} = 1 \end{array}$$

$$\begin{array}{r} 1001 \quad 1011 \\ + \quad \quad \quad 1 \\ \hline 1001 \quad 1100 \end{array} \Rightarrow 9C_{16}$$

D
 $0x56 - 0xAD$
 $(5)_{16} = 0101 = 01010110$
 $(6)_{16} = 0110$

$$\begin{array}{r} 1010110110 \\ - 10101101 \\ \hline 10101001 \end{array} \begin{array}{l} \text{Carry Flag} = 1 \\ \text{Zero Flag} = 0 \\ \text{Auxiliary Flag} = 1 \end{array}$$

A9
 $\Rightarrow A9_{16}$

Module 1 1,2,3

2.

1. Microprocessor have many instructions to move data between memory and CPU
 Microcontroller has one or two instructions to move data between memory and CPU

2. Harvard architecture

3. Flash memory for storing program code
 Data memory for storing data and holding registers
 EEPROM for holding non-volatile data

Gabriel Maxwell
ELE6 3073
Homework 2

4. label

Operation

Operand

Comment

5. The data memory bus only carries data, the program memory bus handles both instructions and data allowing dual data accesses in a single cycle

6. programming Memory - 1K

Data Memory 68 Bytes

SFR default Bank 0

SFR another Bank 0-128 bytes

GPR default Bank 0

GPR another Bank 0-128 bytes

7. programming Memory 7K

Data Memory 256 bytes

SFR default Bank 0

SFR another Bank 0-128 bytes

GPR default Bank 0

GPR another Bank 0-128 bytes

8. programming Memory 14K

Data Memory 256 bytes

SFR default Bank 0

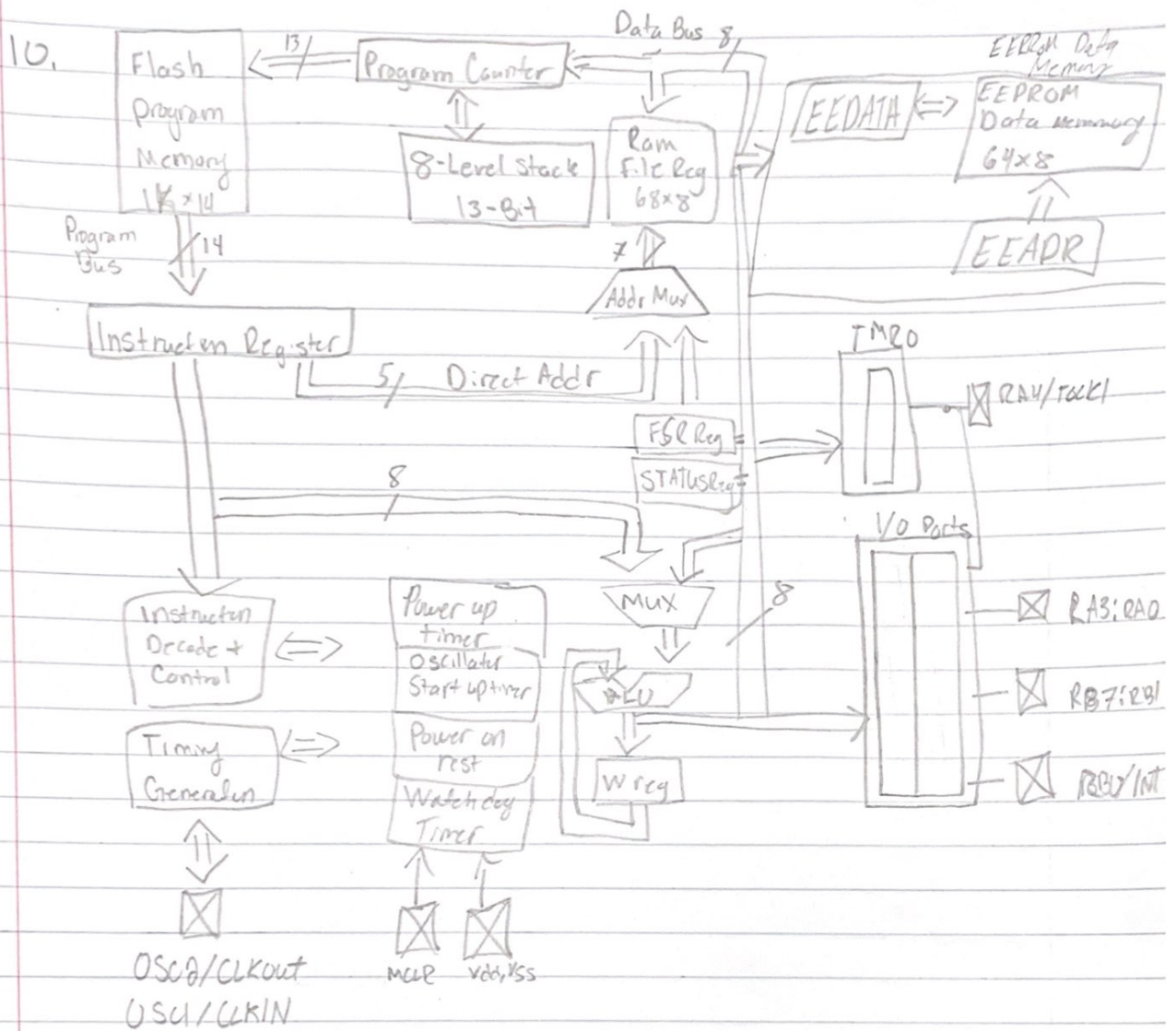
SFR another Bank 0-128 bytes

GPR default Bank 0

GPR another Bank 0-128 bytes

9. .hex extension

Gabriel Maxwell
ELE 3073
Homework 2



PIC16F84A Block Diagram

AVERAGE

ELEG3073 Microprocessor Systems Design

6.5

Homework 2

1. Questions under "Homework" sections of Learning Module #2-1: 2, 3, 4, 5 (b) & (d)

1.1. Convert 7159_{10} to Binary, Octal, and Hex formats.

Decimal	Binary	Octal	Hex
7159_{10}	1101111110111_2	15767_8	$1BF7_{16}$

1.2. Convert $836CE_{16}$ to Binary, Decimal, and Octal formats.

Hex	Binary	Decimal	Octal
$836CE_{16}$	10010011011011001110_2	1210062_2	2233316_8

1.3. Convert 657_8 to Binary, BCD, Decimal, and Hex formats.

Octal	Binary	BCD	Decimal	Hex
657_8	110101111_2	000110101111_{BCD}	431_{10}	$19F_{16}$

1.4. Convert 110001010011001_{BCD} to Binary, Decimal, Octal, and Hex.

Binary Coded Decimal	Binary	Decimal	Octal	Hex
110001010011001_{BCD}	110001010011001_2	50481_{10}	61231_8	6299_{16}

1.5. Perform the following arithmetic operations in Binary and present your results in HEX and also indicate the C & Z & DC flags status after each operation: (NOTE: 0X is another way of indicating that the value is HEX)

1.5.1. (b) $0XEF + 0XAC$

$$0XEF + 0XAC = 11101111_2 + 10101100_2 = 000110011011_2 = 19B_{16} = \boxed{0X19B}$$

1.5.2. (d) $0X56 - 0XAD$

$$0X56 - 0XAD = 01010110_2 - 10101101_2 = 01010111_2 = -57_{16} = \boxed{-0X57}$$

2. Questions under "Homework" sections of Learning Module #1: 1, 2, 3

2.1. State the difference between a microprocessor and a microcontroller?

The *microprocessor* is the “brain” of typical computers and control systems, but it need peripherals interface chips for it cannot function independently.

Microcontrollers are single-chip microcomputers that have peripherals features, core, memory, microprocessors, etc.

2.2. What architecture is the PIC16FXXX based upon?

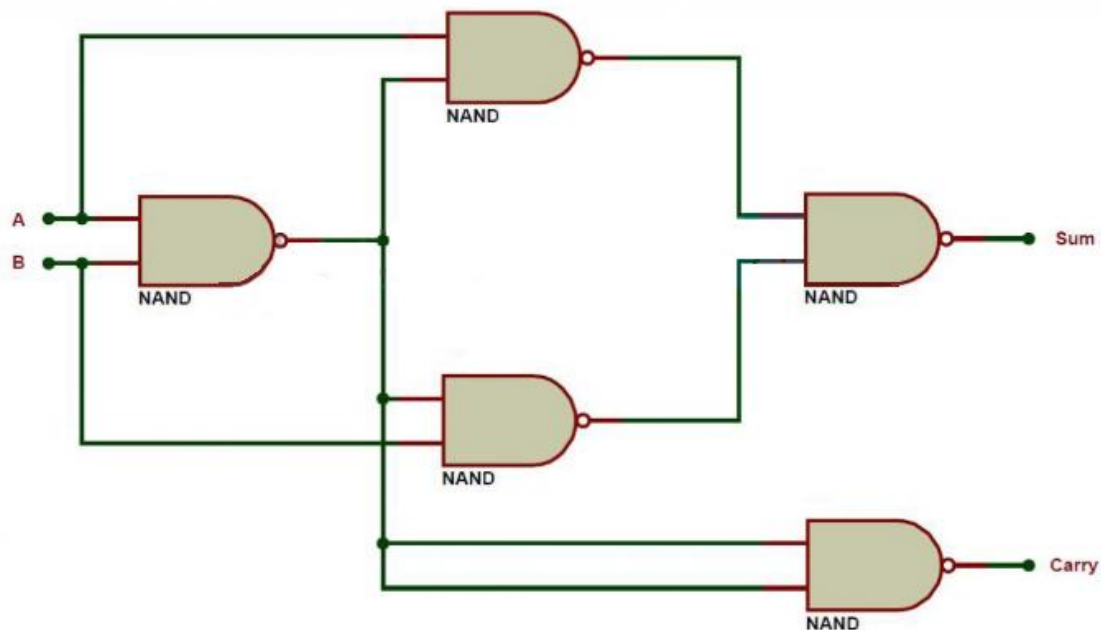
PIC16FXXX is based upon the Harvard Architecture.

2.3. List the types of memory and their uses in the PIC16FXXX family?

While the CPU is getting data, it is also pre-fetching the next program line.

Pipelining allows the Harvard method to be fast, very well suited for control operations, and quite easy to perform I/O operations. In the Harvard Architecture, the program memory and data memory exist as separate spaces (sometimes with different sizes). So, while executing one instruction, the next instruction can be getting fetched.

3. Design a one bit full adder using only one type of logic gate. Draw the truth table, logic expressions, and circuit diagram.



4. State four fields in the Assembly language format.

The four fields in the Assembly language format are label, operation, operand, and comment.

5. State the difference in DM (Data Memory) and PM (Program Memory).

Program Memory is a memory that contains the written program while Data Memory contains special registers.

6. How many bytes are there in PIC16F84A programming memory, data memory? And how many bytes are for SFR and GPR?

In a PIC16F84A, there are 14,000 bytes in programming memory and 512 bytes in data memory. There are about 2944 bytes for file registers.

7. How many bytes are there in PIC16F88 programming memory, data memory? And how many are for SFR and GPR?

In a PIC16F88, there are 56,000 bytes in programming memory and 256 bytes in data memory. There are about 2,944 bytes for file registers.

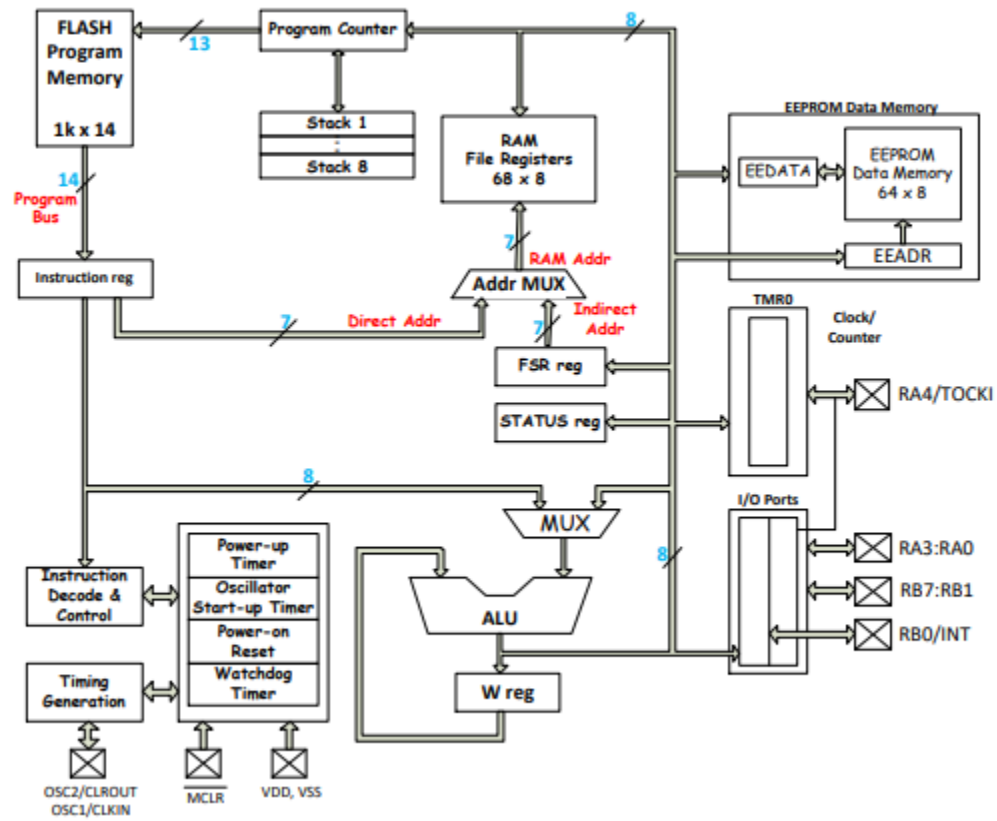
8. How many bytes are there in PIC16F877A programming memory, data memory? And how many are for SFR and GPR?

9. Which file (extension name) is to be download to the PIC16F MCU while using PICKIT2 software?

The .hex file is to be downloaded to PIC16F MCU.

10. Draw the block diagram for PIC 16F84A.

PIC 16F84A Block Diagram



BELOW AVERAGE

Juwan Byers
9/29/20

6

HW #2

1 a) $836CE_{16} \rightarrow 1000\ 0011\ 0110\ 1100\ 1110_2 \rightarrow 2^1+2^2+2^3+2^6+2^7+2^8+2^9+2^{10}+2^{12}+2^{15}+2^{16}$
 $2033316_8 \rightarrow 1076636_{10} \rightarrow 562894_{10}$

b) $657_8 \rightarrow 0010\ 0101\ 0111_2 \rightarrow 1100101011_{BCD}$
 $\rightarrow 110101111_2 \rightarrow 110101111_{BCD} \rightarrow 2^0+2^1+2^2+2^3+2^4+2^5+2^6+2^7$
 $1AF_{16} \rightarrow 862_{10} \rightarrow 431_{10}$

c) $110001010011001_{BCD} \rightarrow 1100010011011_2 \rightarrow 1100011100001_2$
 $(6\ 2\ 9\ 9)_{10} \rightarrow 6299$
 $2^{12}\ 2^{11}\ 2^7\ 2^6\ 2^5\ 2^0 \rightarrow 4026(2^{12})$
 $1100011100001_{10} \rightarrow 18E1_{16}$

1100011100001_{10}

2273

$- 2048(2^{11})$

225
 $- 128(2^7)$

97
 $- 64(2^6)$

33
 $- 32(2^5)$

1
 $- 1(2^0)$
 0

d) $0XEF + 0XAC$
 $14\ 15 \quad 10\ 12$

$1110\ 1111$

$C=1$

$+ 1010\ 1100$

$DC=1$

$11001\ 1011$

$Z=$

After = $0X9B$

ii. $0X56 - 0XAD$
 $10\ 13$

$0101\ 0110$

$- 1010\ 1101$

$1010\ 1101 \rightarrow 0101\ 0110$

$+ 0101\ 0011$

$1010\ 1001$

$2cm$

$0101\ 0011$

$C=0$

$DC=0$

$Z=$

After = $0XA9$

2 a) The difference between a microprocessor and microcontroller is microcontrollers are complete computer systems on a

2 a) (continued)

single chip with the microprocessor as the "brain" of the microcontroller.

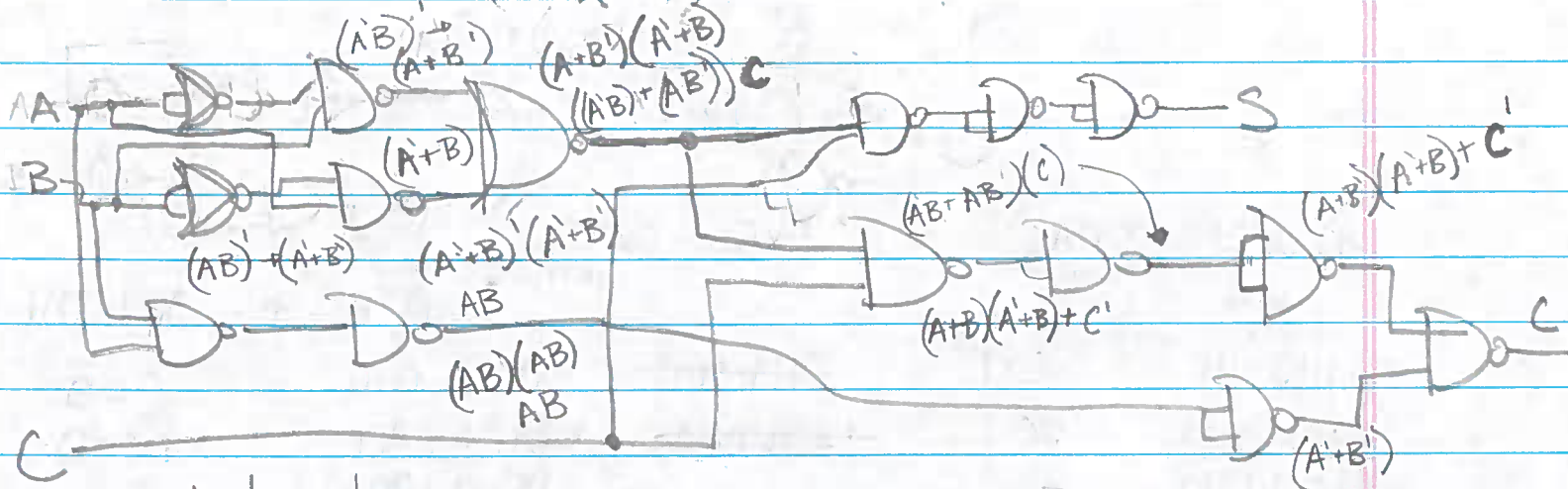
b) PIC16FXXX is based upon Harvard Architecture design.

c) i. ~~84A~~ Memory

(PM) Programming flash memory - holds the certain amount of words for the certain program.

Data Memory - consists of specific function ~~and~~ ^{and} general purpose registers, and EEPROM Data Memory. The SFR are memory locations that reflect the internal architecture and features of the particular device. The GPR are for the programmer's use, for variables needed in the application. The EEPROM are often times used for set points, identification codes, pass codes, hours of operation, mileage, etc.

3. $XOR = \bar{A}B + A\bar{B} \rightarrow (A+B)(A'+B')$



A	B	C	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	0	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	0	1

I tried

X error

$$(A+B)(A'+B')+C(A'+B')$$

$$(A'+AB')C + AB$$

- 4 The four fields in the Assembly language format is label, Operation (OpCode), Operand, and Comment.
- 5 One of the main differences from DM and PM is that PM retains its stored value even after power is removed. It makes me think of how the CD-ROM is able to remember the data file ~~even~~ after it is taken out of the CD player

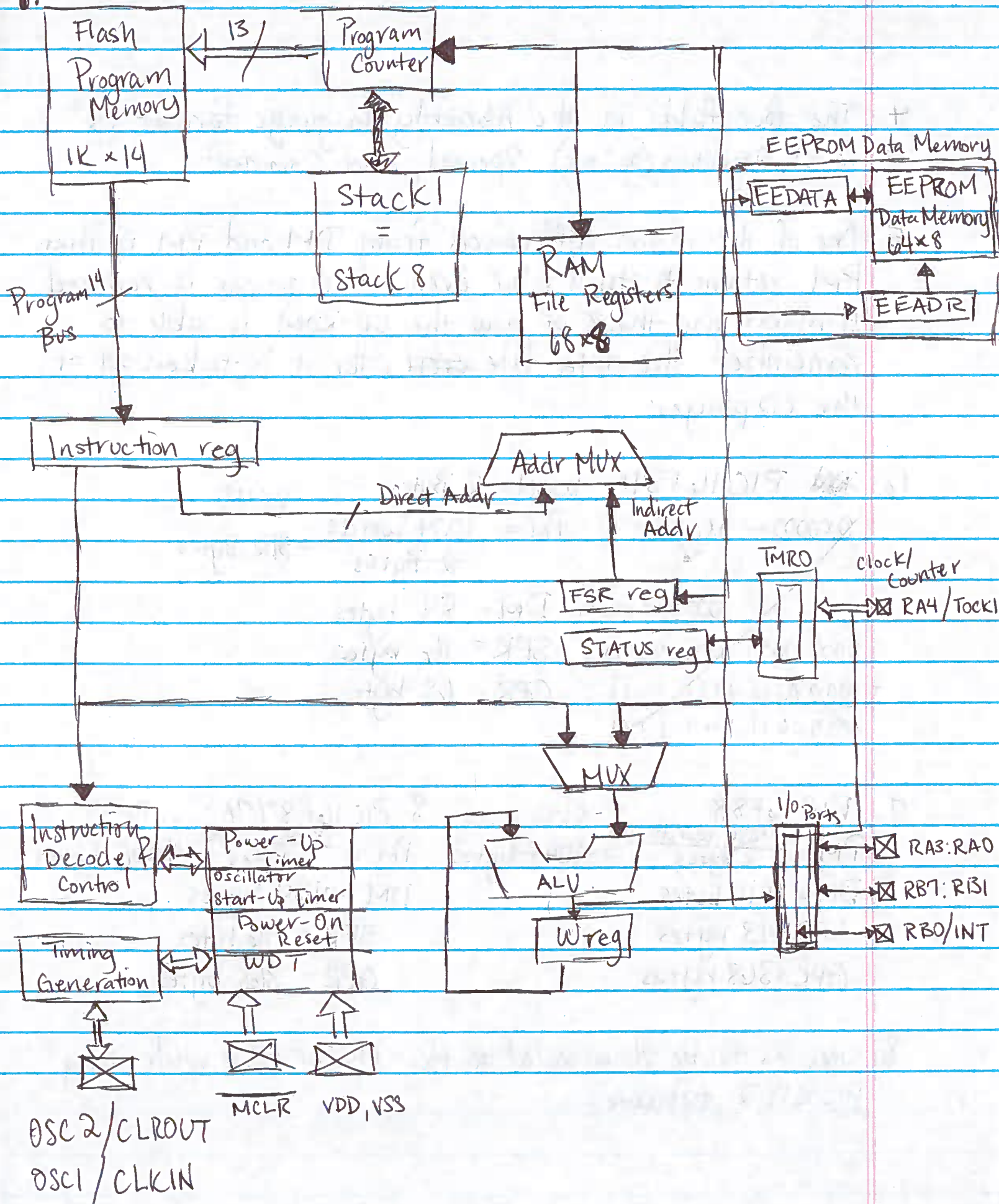
6. ~~PIC16F84A~~ Word = 2 Bytes
~~0x0000 - 0x03FF~~ PM = $\frac{1024 \times 2}{2 \text{ Bytes}} = 1024 \text{ Bytes}$
~~0000 0011 1111 1111~~ DM = 84 bytes
~~0000 0000 0000 0000~~ SFR = 16 bytes
~~+ 0000 0011 1111 1111~~ GPR = 68 bytes
~~0000 0011 1111 1111~~

7. PIC16F88 $\times 2$ 8192 bytes
 PM = $\frac{4096 \text{ words}}{2 \text{ bytes}} = 2048 \text{ bytes}$
 DM = 441 bytes
 SFR = 13 bytes
 GPR = 368 bytes

8. PIC16F877A 16384
 PM = $\frac{8192 \text{ words} \times 2}{2 \text{ bytes}} = 8192 \text{ bytes}$
 DM = 464 bytes
 SFR = 96 bytes
 GPR = 368 bytes

8. .Hex is to be downloaded to the PIC16F MCU while using PICKIT2 software.

9.



**MICROPROCESSOR SYSTEMS DESIGN
FALL 2020**

Title of Assignment: HOMEWORK 3

TOPICS COVERED IN HOMEWORK 3

- 1. Assembly language**
- 2. Flags**
- 3. Watchdog timer**

Homework 3

Solution

1. Perform the following arithmetic operations in binary and present your results in hexadecimal and also indicate the C & Z & DC flags status after each operation. (Note: 0X is another way of indicating that the value is hexadecimal.)

0X3A-0X26
-0X26's 2's complement is:
0010 0110 → 1101 1001 +1 = 1101 1010

0011 1010₂
+ 1101 1010₂
1 0001 0100₂

Result is 0X14

Z = 0, C = 1, DC = 1

2. Explain what will happen after executing the following assembly codes? (Note: Show procedures. Values in hexadecimal of each involved register file or W register. If flag bits are influenced, please also indicate their final values in your answer.)

a. ADDLW 0X0F
(before: W = 0X35)
0011 0101₂ + 0000 1111₂ = 0100 0100₂
0X35 + 0X0F = 0X44
Result in W, all three flag bits are influenced

W = 0X44, DC = 1, C = 0, Z = 0

a. ANDWF TEMP, F
(before: W = 0X18, TEMP = 0X97)
0X18 & 0X97 = 0001 1000 & 1000 0111 = 0001 0000 = 0X10
Result in F → TEMP, only Z is influenced
W = 0X18, TEMP = 0X10, Z = 0

3. If you are using a resonator frequency of 10MHz, please calculate the instruction cycle. If the resonator frequency has been changed to 15MHz, what is the instruction cycle?

Resonator Frequency = 10MHz
Instruction Frequency = (10MHz)/4 = 2.5MHz
Instruction Cycle = 1/(2.5MHz) = 0.4μs

Resonator Frequency = 15MHz
Instruction Frequency = (15MHz)/4 = 3.75MHz
Instruction Cycle = 1/(3.75MHz) = 0.267μs

4. Does the WDT depend on the time supplied by the resonator? Why or why not?

No. The Watch Dog Timer is independent of the time supplied by the resonator because when the system is asleep, the resonator is off. This means there is an internal clock

source which keeps on ticking. WDT depend on an internal RC circuit.

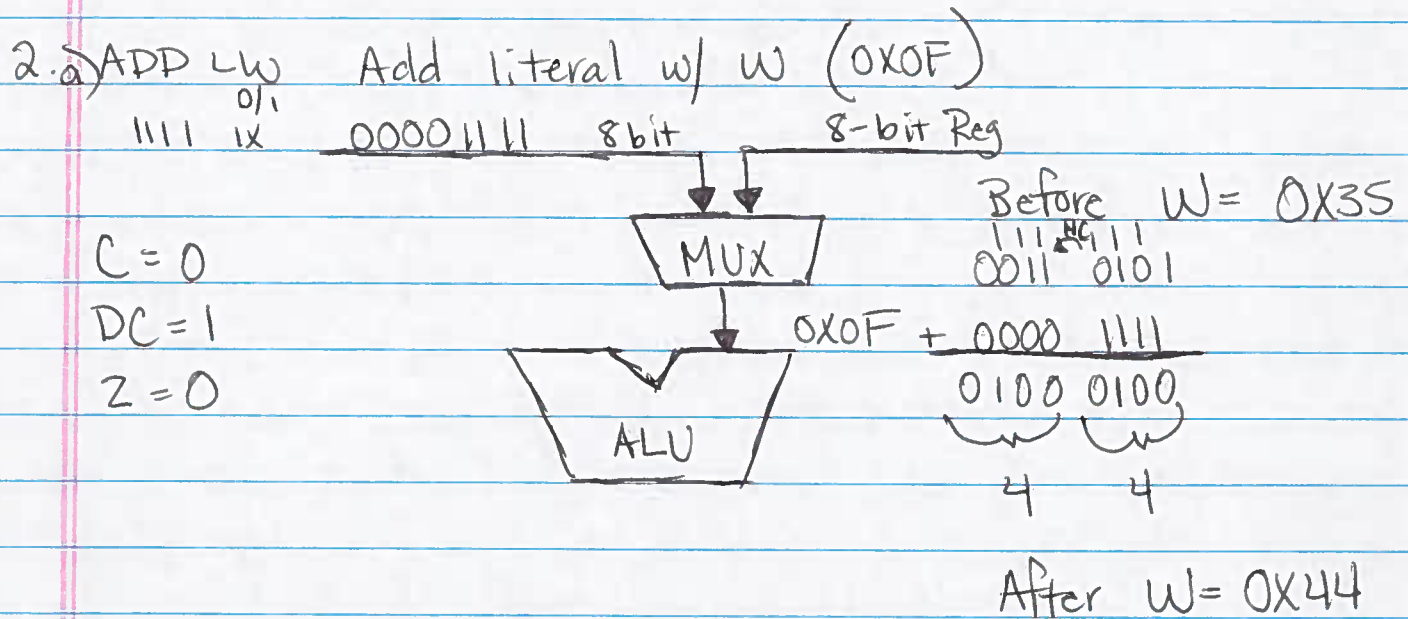
ABOVE AVERAGE

1. $0X3A - 0X26 = 0X14$

$$\begin{array}{r}
 1111 \text{ HC} \\
 0011 \ 1010 \\
 + 1101 \ 1010 \\
 \hline
 110001 \ 0100 \\
 \hline
 1 \ 4
 \end{array}$$

$0011 \ 1010$ $0010 \ 0110$ $2's \ C \ 1101 \ 1010$

$C = 1$
 $DC = 1$
 $Z = 0$



b) ANDWF And W with F Temp, F
before $W = 0X18$, Temp = $0X97$

$$\begin{array}{r}
 0001 \ 1000 \\
 \& \ 1001 \ 0111 \\
 \hline
 0001 \ 0000 \\
 \hline
 1 \ 0
 \end{array}$$

After { $W = 0X10$
Temp = $0X97$

3. Resonator Freq = 10Mhz
Instruction Freq = $\frac{10\text{Mhz}}{4} = 2.5\text{Mhz}$
Instruction Cycle = $\frac{1}{2.5\text{Mhz}} = 0.4\mu\text{s}$
- Resonator Freq = 15 Mhz
Instruction Freq = $\frac{15\text{Mhz}}{4} = 3.75\text{Mhz}$
Instruction Cycle = $\frac{1}{3.75\text{Mhz}} = 0.267\mu\text{s}$
4. The Watch Dog Timer has an independent oscillator. It is not based on the instruction time base.

AVERAGE

HW #3

1. 0X3A-0X26

$$\begin{array}{r} 0011\overset{01}{\cancel{10}}10 \\ - 00100110 \\ \hline 00010100 \end{array}$$

14₁₆ or 0X14

C = 0

DC = 0

Z = 0

② ADDLW 0X0F (before: W = 0X35)

a) 0X0F + 0X35

$$\begin{array}{r} 00001111 \\ + 00110101 \\ \hline 01000100 \end{array}$$

44₁₆ (0X44)

C = 0

DC = 1

Z = 0

b) ANDWF TEMP, F

before: W = 0X18, TEMP = 0X97

- adds contents of W register to TEMP register

$$\begin{array}{r} C = 0 \quad 0X18 - 0001\ 1000 \\ DC = 0 \quad 0X97 - 1001\ 0111 \\ Z = 0 \quad 0001\ 0000 \rightarrow 0X10 \end{array}$$

A	B	S
0	0	0
0	1	0
1	0	0
1	1	1

3. @ 10MHz

$$f = \frac{10}{4} = 2.5 \text{ MHz}$$

$$\frac{1}{2.5} = 0.4 \text{ ns}$$

$$\text{cycle} = 2.5 \times 10^6 \text{ Hz}$$

@ 15MHz

$$f = \frac{15}{4} = 3.7 \text{ MHz}$$

$$\text{cycle} = \frac{1}{3.7 \times 10^6} = 0.27 \mu\text{s}$$

4 The WDT does not depend on the resonator it is independent, because when the system is asleep the resonator is off.

BELOW AVERAGE

Homework 3

Javian Walker 9/28/2020

6

1. Perform the following arithmetic operations in binary and present your results in hexadecimal and also indicate the C & Z & DC flags status after each operation. (Note: 0X is another way of indicating that the value is hexadecimal.)

0X3A-0X26

(On the Back)

2. Explain what will happen after executing the following assembly codes? (Note: Show procedures. Values in hexadecimal of each involved register file or W register. If flag bits are influenced, please also indicate their final values in your answer.)

a. ADDLW 0X0F
(before: W = 0X35)

(On the Back)

a. ANDWF TEMP, F
(before: W = 0X18, TEMP = 0X97)

3. If you are using a resonator frequency of 10MHz, please calculate the instruction cycle. If the resonator frequency has been changed to 15MHz, what is the instruction cycle?

$$\text{Time period of pulse} = \frac{1}{\text{resonator}} \Rightarrow T = \frac{1}{10M} = 0.1\mu\text{sec} \quad \left| \quad \text{if } f = 15\text{MHz}, T = 6.66 \times 10^{-8} \text{sec} \right.$$

$$\text{instruction cycle} = 4 \times T = 0.1 \times 4 = 0.4\mu\text{sec} \quad \left| \quad \text{insturction cycle} = 4 \times T = 0.266\mu\text{sec} \right.$$

4. Does the WDT depend on the time supplied by the resonator? Why or why not?

Yes the WDT depends on the time supplied by the resonator because with it's help, if an error was to occur, the WDT can take time to correct the error.

1.) $0X3A - 0X26$

$X \Rightarrow (3A)_{16} - (26)_{16}$

$X = (3A)_{16} + (2's \text{ complement of } 26)$

Flags:

$C = 1$
 $Z = 0$

$DC = 0$

$$\begin{array}{r} \downarrow \\ \begin{array}{cc} \begin{array}{c} 0010 \\ \hline 2 \end{array} & \begin{array}{c} 0110 \\ \hline 6 \end{array} \end{array} \text{ (Binary)} \\ \begin{array}{cc} \begin{array}{c} 0011 \\ \hline 3 \end{array} & \begin{array}{c} 1010 \\ \hline A \end{array} \end{array} \text{ (Binary)} \end{array}$$

$\Rightarrow 0011 \ 1010 \Rightarrow 3A$

$+ 1101 \ 1010 \Rightarrow 2's \text{ complement of } 26$

$\rightarrow 1) 0001 \ 0100$

End around carry

$\Rightarrow \text{Binary to Hex} \rightarrow \begin{array}{cc} \begin{array}{c} 0001 \\ \hline 1 \end{array} & \begin{array}{c} 0100 \\ \hline 4 \end{array} = 14 \end{array}$

2.) a.) For the instruction $ADDLW \Rightarrow ADDLW K$

The 8-bit literal 'K' value is added to the content of W, and results is stored in W. The flags affected will be N, OV, C, Z and DC.

$\Rightarrow W = 0X35$ before instruction

$\therefore$ After instruction $ADDLW 0X0F \rightarrow W = 0X35 + 0X0F = 0011 \ 0101 = 0X44$

$\therefore$ Here a carry results from the lower 4th bit, so the

status flag $DC = 1$, and $N = 0$, $OV = 0$, $C = 0$, $Z = 0$.

b.) For the instruction $ANDWF \Rightarrow ANDWF f, d$

$\Rightarrow$ AND operation is carried between the content of 'W' and register 'f'. If $d = 0$, result is stored in W and if 1 result is stored back to f (default).

The flags affected by this instruction is N and Z.

$\Rightarrow$ Before instruction $W = 0X18$, $TEMP = 0X97$

$\Rightarrow$ After instruction $ANDWF TEMP, F \rightarrow F = 1111$, W will be the same before and stored in TEMP.

$\Rightarrow 0X18 = 1001 \ 0111$ AND $(0001 \ 1000) = 0001 \ 0000 = 0X10$

$\Rightarrow N = Z = 0$

**MICROPROCESSOR SYSTEMS DESIGN
FALL 2020**

Title of Assignment: HOMEWORK 4

TOPICS COVERED IN HOMEWORK 4

- **Assembly language**
- **Memory**
- **Fundamental components of Cortex processor**

**PRAIRIE VIEW A&M UNIVERSITY
COLLEGE OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
ELEG 3073 Microprocessors Systems Design
Homework 4**

Please see below YouTube links or files to view all the videos of project 1, write a report to evaluate each project (except the one from your group) on:

- (1) Technical content of the project, whether it is related to microprocessor / microcontroller?
- (2) Do the presenters have in-depth investigation on the topic to share with the audience?
- (3) Has the video been set up clearly with visual aids?
- (4) What you've learned from this topic?
- (5) Any suggestions on how to improve?

Upload the report before due date.

There are total of 7 instructional videos, below are 5 Youtube links. I also uploaded two files under this

1. History, architecture, and applications of PIC/Intel/Arduino/
microcontroller/microprocessor
<https://www.youtube.com/watch?v=NZY1TpIh8g4>
2. Arduino
<https://youtu.be/knLuKfnzTHk>
3. AVR
<https://youtu.be/L75qseCDeJE>
4. Van Neumann vs. Harvard
<https://www.youtube.com/watch?v=F-vxKrwLIco>
5. Intel:
https://youtu.be/nEzeQ_IdGGw

ABOVE AVERAGE

1. History, architecture, and applications of PIC/Intel/Arduino/microcontroller/microprocessor
 - 1.1. There was technical content that was related to the arduino board shown in the video but not the PIC microcontroller and the Intel microcontroller.
 - 1.2. The presents seems to have a good knowledge of the topics they covered in there video
 - 1.3. The video was set up somewhat clearly with some visual aid
 - 1.4. I have learned that the PIC microcontrollers and the Intel microcontrollers are used in a variety of applications, applications ranging from cell phones and kitchen appliances to drones and robotics.
 - 1.5. The only suggestion for improvement I would make would be to have better video quality.
2. AVR
 - 2.1. There were not muc technical content in the video
 - 2.2. The presenter seems to have done service level investigation on the topic
 - 2.3. There is not much visual aids in the video
 - 2.4. I have learned AVR using the Harvard computer structure.
 - 2.5. I would suggest incorporating more visuals aid like a picture of the actual microcontroller.
3. Van Neumann vs. Harvard
 - 3.1. There was some technical content shared through audio in the video but not visually.
 - 3.2. It seems that the presenter has an in depth investigation and understanding of the topic
 - 3.3. This video has no visual aids at all
 - 3.4. I did not learn anything new that I already didn't know
 - 3.5. I would suggest to use more visual aids next time
4. Intel
 - 4.1. There were a substantial amount of technical content that was discussed in the video
 - 4.2. It appears that the presenter has done a good amount of investigation and research on the topic he is presenting.
 - 4.3. This video has been set up clearly but it could use more visual aids.
 - 4.4. I learned the history of the Intel microcontroller chips
 - 4.5. I would suggest to use more diagrams of the chips and drawings of the block diagrams of the chips

AVERAGE

EVALUATION REPORT

Homework #4



NOLAN COSTON
DR. CUI, SUXIA
OCTOBER 25, 2020
PRAIRIE VIEW A&M UNIVERSITY

History, Architecture and Applications of Intel, PIC, and Arduino Microcontrollers/Microprocessors

The content of the project related to microprocessors and microcontrollers is good. The presenters explain great details and facts about the Intel, PIC, and Arduino microprocessors and microcontrollers. The video's visual feed has been blurry for a minute, but the rest the runtime has fair quality. From these subjects, I have learned few things about Intel and Arduino that I did not know before.

Arduino

This group is giving information about the Arduino board. The presenters do explain good details about Arduino with a great outside setting in the background, however it has been slightly difficult trying to listen the presenters' voices over the sound of vehicles passing by. From what I have heard, the Arduino board holds a single chip microcontroller, known as AVR, that has been created by the Atmel Corporation, and I find this information pretty interesting that Arduino uses Atmel technology. I have also learned that there was more than one Arduino board that was created.

Van Neumann v. Harvard Architecture

This project talks about the Van Neumann and Harvard Architectures. The presenter explains the history and difference between the two architectures. A picture that illustrates the idea of both architectures shows in the beginning the video but fades away before the speaker begins speaking. Even though, there was blue screen throughout the whole video, I heard the speaker present the genius behind the sophisticated Van Neumann architecture and the MORE sophisticated Harvard architecture.

Intel

This project presents about Intel and the marketing on anything with an Intel microprocessor chip. The presenter does give an in-depth investigation on Intel that is shared with the audience. The video has good visual of the presenter and great audio. From what the speaker presented in the presentation, I have learned that many microprocessor chips are not too expensive and that many products use Intel microprocessor chips.

BELOW AVERAGE

- (1) Technical content of project, whether it is related
- (2) Do presenters have in-depth investigation on the topic to share
- (3) Has the video been set up clearly?
- (4) What have you learned?
- (5) Any suggestion on how to improve?

6

Arduino

- ① The video is indeed very technical and talk about microcontroller.
- ② The video is indeed in-depth contains: specs, history, background,
- ③ The video is very clear and audio too. and importance.
- ④ First board was made in 2005, in Italy. Usually partnered with AVR.
- ⑤ Include more information in regards to the Also, Arduino practical use of the controller/board. In regards has tons to "Why is it important to students?" of variations.

AVR

- ① The video is technical it talks about the AVR microcontroller.
- ② The video is in-depth goes into the history, background, specs, and also breaks down some software.
- ③ The video is clear and audio too.
- ④ These controllers can be found in most of our everyday appliances, as well as the many features of microcontrollers, storage, clocks, etc.
- ⑤ Commit some of the information to memory, don't look like you're reading.

Van Neumann vs. Harvard

- ① The video is technical it talks about the different memory processors.
- ② The video is in-depth and states the many differences between the two.
- ③ The video is somewhat clear.
- ④ Von Neumann was the first processor, and Harvard is the more utilized one.
- ⑤ There were a lot of pauses, but still good information.

Intel

- ① The video is technical it talks about the Intel processor.
- ② Yes
- ③ yes
- ④ I learned that the chips took of in the P100's, there was a big need for them, because of this there was a large demand for bigger and better.
- ⑤ Video was good as well as the information.

Intel - O'Mara

- ① Very technical good information on intel microprocessor.
- ② yes
- ③ yes
- ④ I learned more about the history and what made intel a leader in microprocessors during the time.
- ⑤ Good video and wonderful information.

Avdio - Marvin

- ① The video is indeed technical.
- ② Yes
- ③ yes
- ④ I learned that in terms of intro microcontrollers / processors Avdio is amongst the best, but doesn't compete well beyond the basics.
- ⑤ Great video presentation, as well as live presentation of parts.

**MICROPROCESSOR SYSTEMS DESIGN
FALL 2020**

Title of Assignment: HOMEWORK 5

TOPICS COVERED IN HOMEWORK 5

- 1. Knowledge of microcontrollers**
- 2. Engineering applications**

Homework 5 Solution

Textbook Chapter 6 questions 3 & 5.

Translate assembly language into machine language: BCF STATUS, Z

01 0001 1000 0000

Because the format is BCF is: 01 00bb bfff ffff

The operation code of BCF is: 01 00

STATUS register is: bbb = 0x03 = 011

Z bit is at bit0: fff ffff = 000 0000

3. The Timer 0 module of a 16F84A is initiated by loading its Control SFR with value 00000011. The oscillator frequency is 8MHz.

- a) Under these conditions, what is the frequency at the input to the Timer?
- b) If the counter is initially cleared to zero, how long does it take before it first overflows.
- c) If INTCON was initially cleared to 0, what is its value immediately after this overflow occurs?

The Timer is using the internal instruction cycle clock, prescaler is assigned to Timer 0, at a value of 1:16.

- a) The instruction cycle frequency is 2MHz ($f_{osc}/4$). This is further divided by 16, giving an input frequency to the Timer of 125kHz.
- b) The period of 125kHz is 8 μ s. There will be 256 cycles before overflow, i.e. 2.048ms.
- c) 0000 0100. The flag is set even if the interrupt is not enabled.

5. In an application using the 16F84A, a regular timed interrupt is required. The clock oscillator frequency is 4MHz, and an interrupt frequency in the region of every 2ms is required. Describe how you would configure the TMR0 module.

- ii) The internal clock frequency will now be 1MHz. The interrupt will occur every 256 cycles of the frequency presented to the counter. Without further division this is every 256 μ s. If a division of 8 occurs, using the pre-scaler, an interrupt period of (8x256 μ s) i.e. 2.048ms, occurs, which meets the need of the question. An appropriate Option register setting to achieve this will be XX0X 0010.

ABOVE AVERAGE

Gabriel Maxwell

11/22/2020

Chapter 6 Homework

Homework 5

3. Oscillation frequency = 8 MHz option register = 00000011

a). pic16f84A force = $\frac{\text{force}}{4} = \frac{8M}{4} = 2MHz$

ps2	ps1	ps0	
0	1	1	so 1:16 is selected

The Frequency input to Timer 0 is

$$\frac{2M}{16} = 125 KHz$$

b) starting value = 0

final value = 255 (8 b. timer)

time per instruction = $\frac{1}{125 \times 10^3} = 8 \mu s$

$$\begin{aligned} \text{the total time} &= 8 \mu s \times 256 \\ &= 2.048 ms \end{aligned}$$
c) INTCON = 08 after overflows
Because TOIF = 1

5 the force = 4 MHz

$$T_{osc} = \frac{4}{4M} = 1 \mu s$$

Select pre scalar 1:8

ps2	ps1	ps0
0	1	0

option register value = 00000010

time per instruction = $1 \times 8 \times 10^{-6} = 8 \mu s$ To get 2ms time value = $\frac{2m}{8\mu} = 250$

So TMRO is loaded with 256 - 250 = (06)

→ TMRO = 06

INTCON is loaded with (10100000)₂ = (A0)₁₆

Translate assembly language into machine language: BCF STATUS, Z

Step 1

BCF STATUS, Z

operands $0 \leq Z \leq 127$ $0 \leq b \leq 7$

operation: $0 \rightarrow Z$

Status Affected: None Encoding: 01 00bb bZZZ ZZZZ

Description: Bit 'b' in register 'Z' is cleared

Words: 1 Cycle: 1 Q Cycle

Activity: Q1 Q2 Q3 Q4

Decode Read register 'Z'

process data Write register 'Z'

Step 2

Before instruction execution $W = 0x17$

$FSR = 0xc0$

Register at address (FSR) contains the value $0x02$

After instruction: $W = 0x17$

$FSR = 0xc2$

Register at address (FSR) contains the value $0x07$

BCF is a Resolving Command Instruction used

Answer ~~01000100000011~~

AVERAGE

6

3 (a) For PIC16F84A $f_{osc} = \frac{f_{osc}}{4} = \frac{8M}{4} = 2MHz$

PS2 PS1 PS0
0 1 1

$\Rightarrow$ pre scalar 1/16 is selected

$$\frac{2M}{16} = \boxed{125 kHz}$$

(b) Time per instruction = $\frac{1}{1.25 \times 10^3} = 8\mu s$

$$\text{Total time} = 8\mu s \times 256 = \boxed{2.048 ms}$$

(c) INTCON = 05 after overflows
blk TOIF = 1

5

$$f_{osc} = 4MHz$$

$$T_{osc} = \frac{1}{4M} = 1\mu s$$

Select prescaler 1/4

$\therefore$ PS2 PS1 PS0
0 1 0

Register value .. 000 000 10

$$\text{Time per instruction} = 1.8 \times 10^{-6} = 8\mu s$$

$$\frac{2M}{8\mu s} = 250$$

$$TMRD : 256 - 250 = (6) \Rightarrow TMRD = \boxed{06}$$

$$INTCON : (10100000)_2 = (A0)_{16}$$

BELOW AVERAGE

3) The Timer 0 module of a 16F84A is initiated by loading the **OPTION** register with value 00000011. The oscillator frequency is 8 MHz

(a) Frequency = **124MHz**

(b) Initial Value = 0

Final Value = 255

Time per instruction = $(1 / (125 \times 10^3)) = 8\mu\text{s}$

Total time = $(8\mu\text{s} \times 256) = \mathbf{2.048ms}$

(c) **INTCON** = 08 after overflow because TOIF = 1

5) $f_{\text{osc}} = 4\text{MHz}$

$T_{\text{osc}} = (4 / 4\text{MHz}) = 1\mu\text{s}$

Time per instruction = $8\mu\text{s}$

Value of 2ms of time = $(2\text{ms} / 8\mu\text{s}) = 250$

So TMR0 is loaded with $256 - 250 = 06$ **TMR0 = 06**

INTCON is loaded with $(10100000)_B = (A0)_H$

MICROPROCESSOR SYSTEMS DESIGN FALL 2020

Title of Assignment: MIDTERM PROJECT

TOPICS COVERED IN MIDTERM PROJECT

- 1. Microcontroller applications**
- 2. Different microcontrollers**
- 3. History of microprocessor and microcontrollers**

**PRAIRIE VIEW A&M UNIVERSITY
COLLEGE OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
ELEG 3073 Microprocessors Systems Design**

**Project 1
(Midterm Project)**

In this mandatory project, students can work as a group of maximum 2 students toward a microcontroller/microprocessor related topic. It can be hardware or software related. You can choose one microcontroller, such as PIC, ARM, AVR, Intel, Arduino, or etc. Introduce the history of development, the advantages and disadvantages, popular products in current market, or research in academia. The project is due on Tuesday of week 8, and by week 6, each group needs to send instructor the names of your group members and the tentative project title. Each group needs to turn in one group written report of at least three pages long (not including cover page) with single space and 12 font. The following format should be observed for preparation of your written reports.

1. Cover Page: indicating title and all partners
2. Introduction/Overview
3. Theory (the architecture of the microcontroller/microprocessor) and application of this knowledge
4. Discussion of methodology leading to advance of microcontroller/microprocessor techniques
5. Discussion/Explanation of how the applications influence microcontroller/microprocessor industry
6. Conclusion
7. References

Each group is also required to provide an instructional video of less than 10 minutes.

Your project will be evaluated as a group with the rubric:

Cover page _____ (1 points)

Quality of visuals and or written materials _____ (6 points)
(i.e readable text, no typos)

Demonstrated knowledge of research/ application areas _____ (6 points)

References _____ (1 points)

Quality of presentation delivery, video organization & clarity) _____ (6 points)

TOTAL _____ (20 points)

ABOVE AVERAGE

18

Arduino: The beauty of on board storage with a
microcontroller

Juwan Byers (22017151)

Dr. Cui

10/12/2020

Introduction

The Arduino is an open source hardware concept that was developed by Massimo Banzi, David Cuatrilles, Tom Igoe, Gianluca Martino, and David Mellis. It was made to be an easy-to-use programmable device primarily for interactive art design projects. The AVR family of devices was one of the first to incorporate on-board flash memory for program storage, instead of the one-time programmable ROM (read-only memory), EPROM (erasable programmable read-only memory), or EEPROM (electrically erasable programmable read-only memory) found on other microcontrollers (Hughes, 2016). In the document, there will be a discussion of the methodology and creation of the Arduino, The influence that it has on the microcontroller/microprocessor industry, and the theory and application of the device.

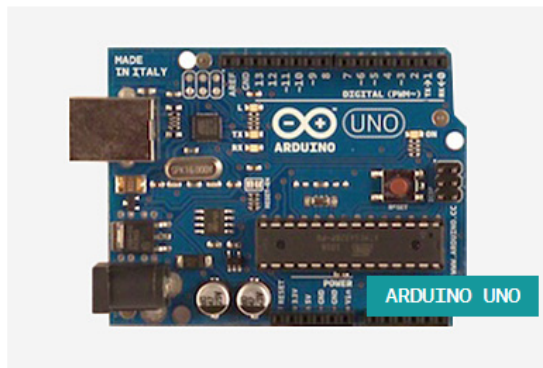
Theory

The Arduino family of boards use processors developed by the Atmel Corporation of San Jose, California. Most of the Arduino designs utilize the 8-bit AVR series of microcontrollers, with the Due being the primary exception with its ARM Cortex-M3 32-bit processor (Hughes, 2016). The Arduino board for all of the boards in the Arduino family is a basic Atmel AVR development board with the Arduino software; the Arduino software is what sets the Arduino apart from all of the other microcontroller/microprocessors. As far as the internal architecture of the device, the difference between AVR microcontrollers depends on the amount of on-board flash memory and the available I/O functions. A list of some of the essential features of the AVR microcontrollers include RISC architecture, one hundred thirty one instructions, thirty two eight bit general purpose registers, up to twenty megahertz clock rate, on-board memory, flash program memory (up to two hundred fifty six kilo), on-board EEPROM (up to four thousand kilo), Internal SRAM (up to thirty two thousand kilo), Operating voltage $VCC = 1.8$ to 5.5 volts DC (Hughes, 2016). As mentioned in the introduction, the memory of the AVR has the ability to be stored onto the board; it uses flash and EEPROM for the storage of data on the board. The flash and EEPROM can be loaded externally, and both will retain their contents when the AVR is powered off. The SRAM is volatile, and its contents will be lost when the AVR loses power (Hughes, 2016). The framework for the inner working of the microcontroller are below in figure one and two. Figure two is the framework of one of the more advanced Arduino models.

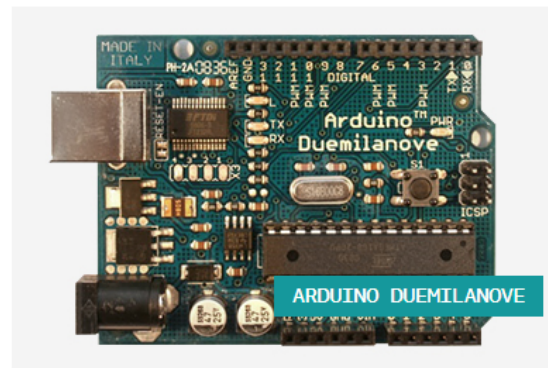
For the function of each Arduino board, they have the AVR board with the microcontroller and the IDE software. The software is compatible with C and C++. It doesn't have to be used with a specific computer when the board is programmed because it can be used with Mac, IOS, and LINUX.

Advancement of Techniques

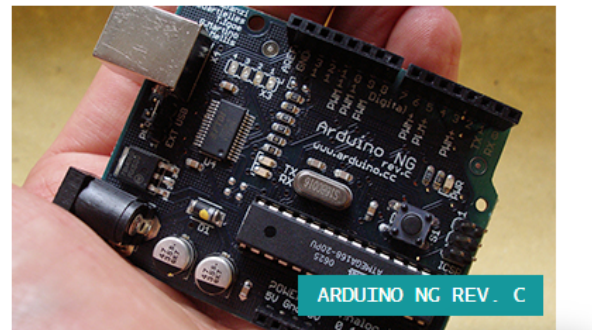
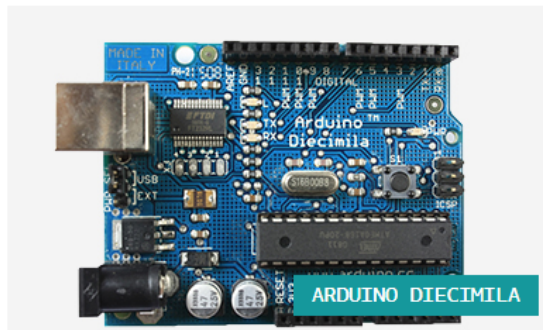
With the Arduino, you can design and build devices that can interact with its surroundings (Anna, 2020). The Arduino boards are basically a tool for controlling electronics (Anna, 2020). The Arduino is continually improved every year and mainly has the RAM and flash improved on their newer models such as the Due that comes out every year. Here is a picture of some of the new Arduinos that have just been released.



This is the latest revision of the basic Arduino USB board. It connects to the computer with a standard USB cable and contains everything else you need to program and use the board. It can be extended with a variety of shields: custom daughter-boards with specific features. It is similar to the Duemilanove, but has a different USB-to-serial chip the ATmega8U2, and newly designed labeling to make inputs and outputs easier to identify. [More](#)



The Duemilanove automatically selects the appropriate power supply (USB or external power), eliminating the need for the power selection jumper found on previous boards. It also adds an easiest to cut trace for disabling the auto-reset, along with a solder jumper for re-enabling it. Note: around March 1st, 2009, the Duemilanove started to ship with the ATmega328P instead of the ATmega168. [More](#)

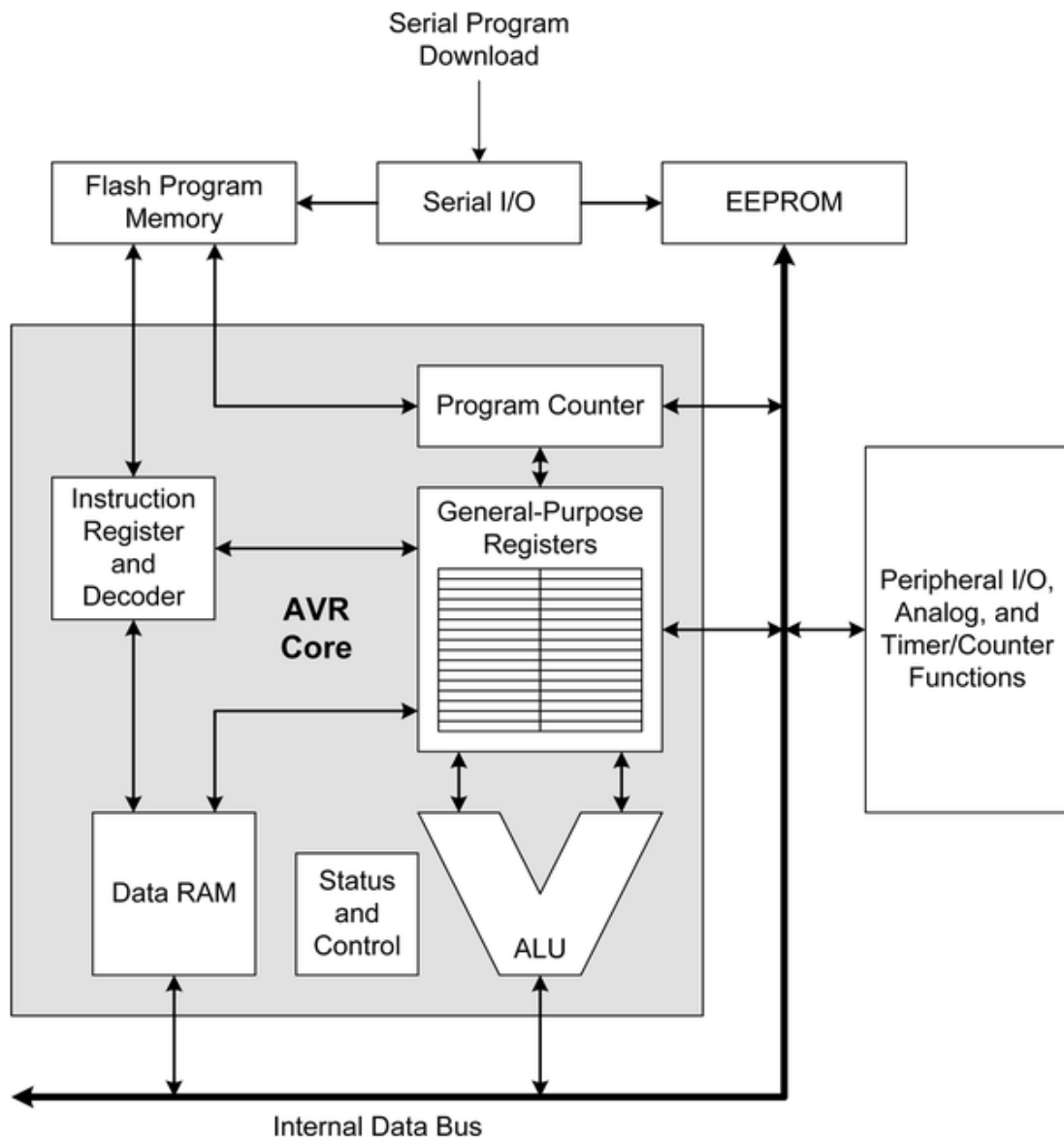


Impact on Industry

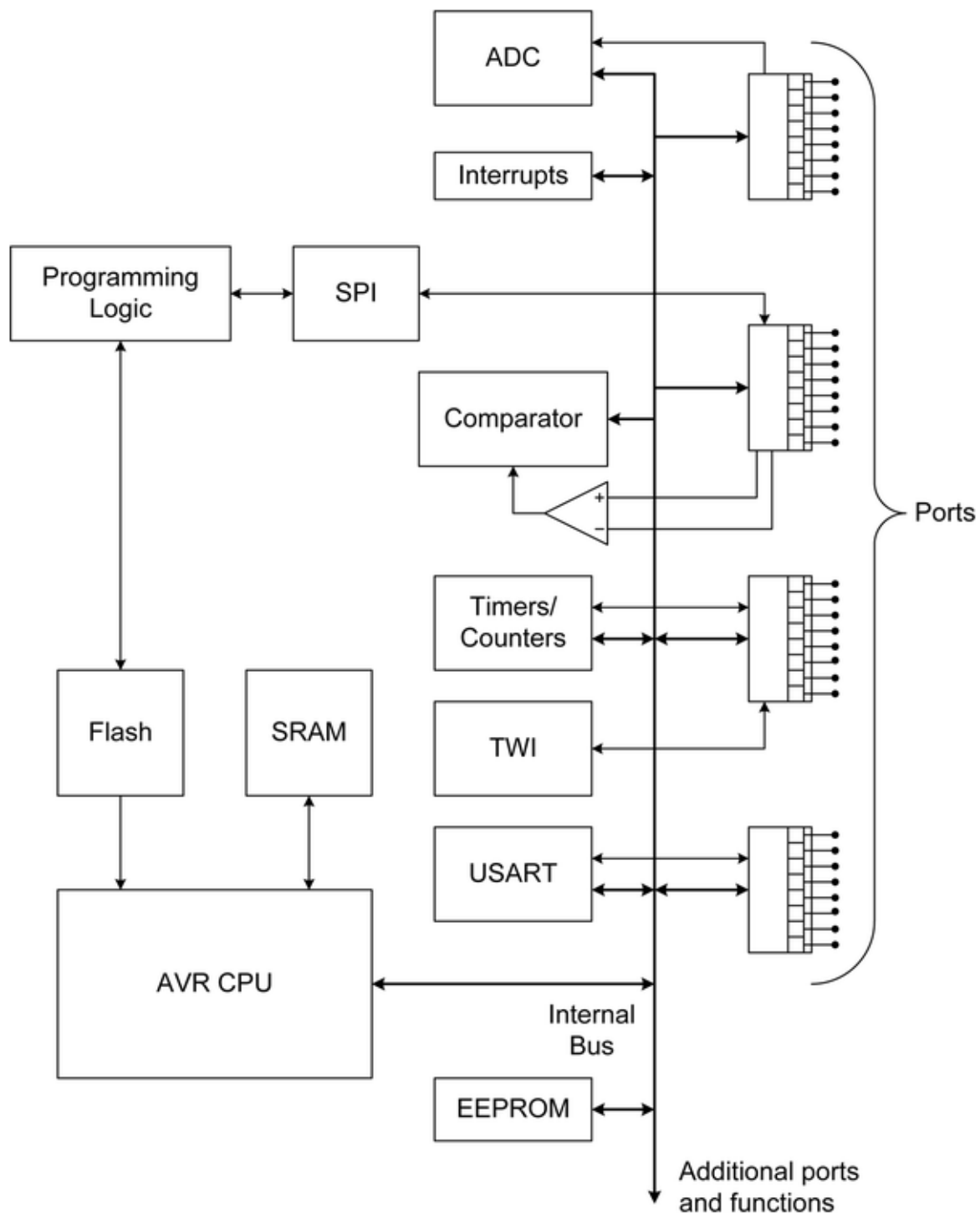
The Arduino allows for many tasks to be completed in certain robots of use such as a remote control lawnmower. They can be used for the automotive industry. For example, Bill Wong used the Arduino Uno to display the speed of his vehicle on his driver side window so that he did not have to look away from the road (Sanchez, 2017). Another user by the name of ChrisJohnson55 shows that with the use of the Arduino microcontroller he was able to use the remote start feature through his laptop using Bluetooth (Sanchez, 2017).

Conclusion

The Arduino is a very simple RISC that allows for certain jobs to be completed. It can be used universally whether it is placed in a car for the sensor lights or even to turn on the ignition using the Bluetooth. The advancement of it will continue as there will be more I/O inputs and memory that will be able to be programmed on the board in the future. The advancement of the device will lead to more complex jobs that can be done with a single microcontroller.



AVR CPU block Diagram



References

- Anna, N. (2020, September 30). Arduino Uno vs Mega vs Due: The Differences. Retrieved October 13, 2020, from <https://all3dp.com/2/arduino-mega-vs-uno-vs-due-differences/>
- Hughes, J. (2016). *Arduino: A Technical reference*. O'Reilly Media.
- Sanchez, A. (2017, May 26). 5 Arduino projects to upgrade your car. Retrieved October 12, 2020, from <https://opensource.com/article/17/5/upgrade-your-car-these-5-arduino-projects>

AVERAGE

**History, architecture, and applications of PIC/Intel/Arduino/
microcontroller/microprocessor**

**Omer Adnan-Intel
George Polk-PIC
Marvin Berrios-Arduino**

Introduction:

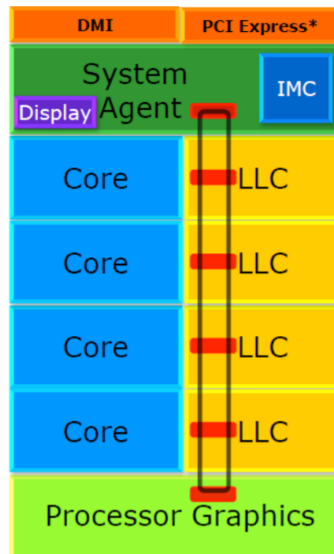
Microcontrollers and microprocessors are a prominent part of modern society. Microcontrollers are compact integrated circuits that perform specific operations. Microcontrollers include RAM (random access memory), ROM (read only memory) and other peripherals including the CPU or processor. Microprocessors also consist of a CPU and need other peripherals to do a task. Laptop and PCs are good examples of where microprocessors are used. Due to massive use of microcontrollers/microprocessors, there are several companies that manufacture and include them into their products. This report covers the history, architecture and applications of Intel, Arduino and Raspberry Pi microcontrollers and microprocessors.

Intel Microprocessors**The architecture of the Intel microprocessors and application of this knowledge:**

One of the most well known microprocessor brands is Intel. Their microprocessors are used in the majority of computers today. Intel's x86 architecture has become a staple among most computers. Intel's x86 architecture was designed in 1978, and was one of the first architectures for microprocessor-based computing. Key features of the x86 include providing a logical framework for executing instructions through a processor allowing softwares and instructions to run on any processor in the Intel 8086 family and also providing procedures for utilizing and managing the hardware components of the CPU. The x86 architecture primarily handles programmatic functions and provides services, such as memory addressing, software and hardware interrupt handling, data type, registers and input/output (I/O) management. Classified by bit amount, the x86 architecture is applied in multiple microprocessors, including 8086, 80286, 80386, Core 2, Atoms and even more newer processors like the i5 and i7. Additionally, other microprocessor manufacturers, like AMD and VIA Technologies, have adopted the x86 architecture.

Discussion of methodology leading to advance of microcontroller/microprocessor techniques:

Every branch of the Intel architecture or x86 contains the same basic features and functionality as chips from previous generations while also retaining backward compatibility with them. Aside from that each new generation adds new features and functionality to the mix. Intel Pentium processors for example, added multimedia extensions that vastly increased audio and video processing. Floating-point units (FPUs) that perform operations on floating point numbers became a standard feature of Intel architecture. Today encryption and decryption extensions, power-management features, and multilevel caches are now found on most Intel architecture processors. Data paths have widened from 8 bits to 32 bits, 64 bits, and even 128 bits. Operating frequencies have jumped from a few megahertz to 2 GHz (two billion cycles per second) and beyond. Many modern intel architecture chips include dual, quad, octa and even 12 core processors that operate within a single chip. Some include a feature called multithreading that allows a single core to perform multiple tasks. The figure below shows a 64 bit architecture called the intel i7.



As seen in the block diagram the architecture uses 4 cores, a two channel DDR3 DRAM controller, integrated graphics and shared caches. It also showcases the PCIe lane. PCIe interface is the fastest I/O interface in the intel architecture. PCIe supports higher transfer rates. The original PCIe specification specifies data rates of 2.5 GT/sec (250 MB/sec) per lane, the same as DMI. The second generation of PCIe doubles the data rate to 5 GT/sec (500 MB/sec). Due to the advancement of these technologies, the processors are able to operational frequency of the systems, thereby increasing power and efficiency.

Discussion/Explanation of how the applications influence microcontroller/microprocessor industry:

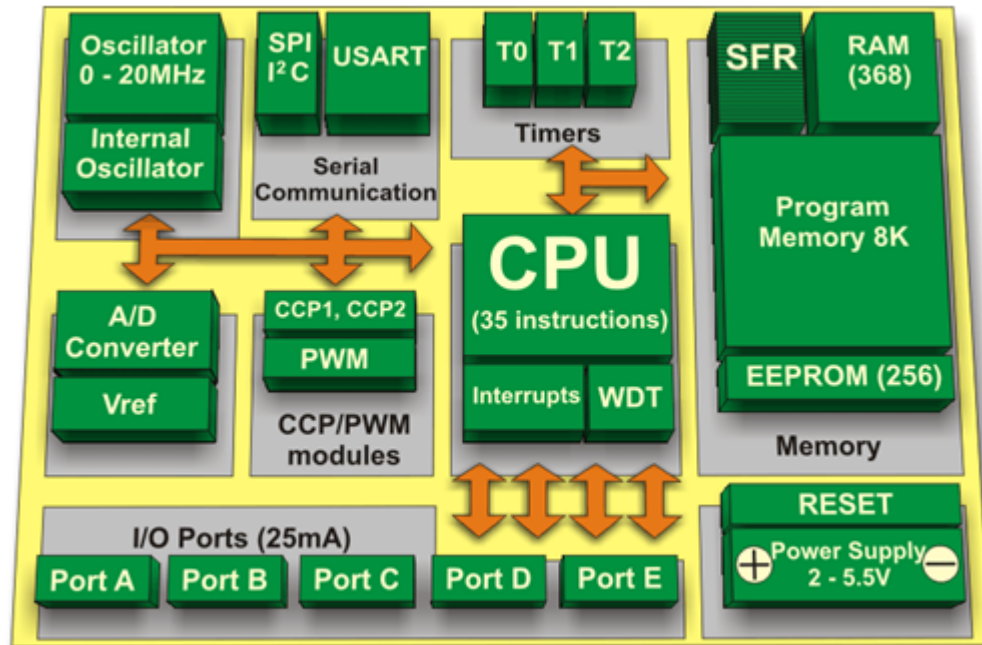
Intel's microprocessors have left a huge mark on the computer and technology industry. Even rival companies like AMD have used the x86 architecture designed by intel. Aside from computers intel microprocessors are used in virtually every field of electronics, including automotive, industrial, automation, robotics, consumer electronics, image processing, networking, encryption, military, construction, medical, energy, and other industries. The heavy application of intel's microprocessors in the tech industry has even impacted the supply and demand of the processors. Since 2018, Intel has been trying to resolve supply shortage problems by investing in their manufacturing capacity, but even then they are struggling to meet the rising demand. This shows how a microprocessor architecture can cause problems in the industry due to the heavy requirements of processors in the modern world.

PIC

History/ Background - The PIC microcontroller was developed in the year 1993 by microchip technology. PIC stands for Peripheral Interface Controller. It was initially developed for supporting PDP(Programmed Data Processor) computers to control its peripheral devices, and therefore, named as a Peripheral Interface Controller. These microcontrollers are very fast and it's easier to execute a program compared with other microcontrollers. PIC microcontrollers are very popular due to their ease of programming, wide availability, easier interfacing with other peripherals, low cost, large user base, and serial programming capability.

Architecture - The PIC microcontroller architecture is comprised of CPU, I/O ports, memory

organization, A/D converter, timers/counters, interrupts, serial communication, oscillator and CCP module. In the figure below you can see the basic architecture of a PIC.



Architecture of PIC Microcontroller

Application in Industry - At present PIC microcontrollers are extensively used for industrial purposes due to low power consumption, high-performance ability, and ease of availability of its supporting hardware and software tools like compilers, debuggers, and simulators. PIC microcontrollers are found in most electronic devices such as alarm systems, traffic control systems and RFID based security systems.

Discussion of methodology: What led to the creation of the PIC was the need for an affordable Microcontroller there are several advantages to the PIC controller that make it so much more commonly used.

Advantages:

1. PIC microcontrollers are more consistent and have less faults. The performance of the PIC microcontroller is very fast.
2. The power consumption is less than other microcontrollers and programming is also much easier.
3. Interfacing of an analog device is easy without any extra circuitry.

Disadvantages:

- The Peripheral Interface Controller has only one accumulator.

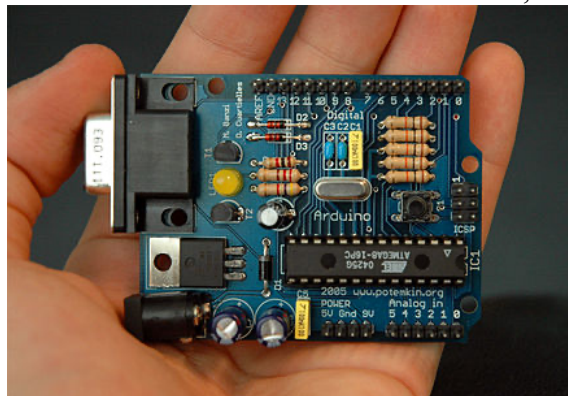
- Small instruction set.
- Register banking switch required to access RAM of other devices.
- Operations and registers are not orthogonal.
- Program memory is not accessible.

Advances: The only difference between the 16F84 and the newer PICs is that they are eight times more than the code size, with more RAM and much more I/O pins, a UART, A/D converter and a lot more features.

Arduino

History/Background: The Arduino project was started at the Interaction Design Institute in Ivrea, Italy. In 2003, Hernando Barragan created the development platform ‘Wiring.’ In 2005, Massimo Banzi, David Mellis, David Cuartielles, and two others extended ‘Wiring’ by adding support for the cheaper ATmega8 microcontroller, which was then called Arduino. This is where Arduino took off. In 2008, the five co-founders of the Arduino project created a company, Arduino LLC, to hold trademarks associated with Arduino.

Architecture/Definition: Early Arduino boards came with an RS-232 serial interface and an Atmel ATmega8 microcontroller chip. It had 14 digital input/output pins and 6 analog input pins, and the power connector at the lower left. Newer boards are similar, but have more features.



Arduino is an open-source electronics platform based on easy to use hardware and software, it is a development board which contains a microcontroller mainly 8-bit AVR. It is designed to make electronics more accessible to artists, designers, and anyone interested in creating interactive environments or objects. The Arduino IDE, is the open source software, the coding is easy to write and upload to the board. It runs on Windows, Linux, and Mac OS X. It can be used with any type

of Arduino board.

Advantages: Arduino products are very cheap compared to other more complex and expensive microcontroller-based boards. The hardware is easy to use and so is the software, Arduino IDE.

Disadvantages: Even though Arduino is very useful for various purposes, it cannot do too complex functions due to its limited hardware. Arduino is more of a beginner's guide into electronics and basic projects dealing with microcontrollers. For advanced projects Arduino would probably be useful in initial phases but not for completion stages.

Academia Research Products: Arduino Products include but are not limited to: ATmega8, ATmega168, ATmega368, ATmega1280.

Applications in Industry: Arduino products have helped with the development of microcontrollers and the importance of their use since 2003. This platform can be used by people of all ages and all levels of experience. This is what makes Arduino so important in terms of applications in the microcontroller industry.

Conclusion:

This report shows the impact of microcontrollers and microprocessors in the modern society. Some of the few examples it covers are Intel, PIC and Arduino. Intel is one the oldest microprocessing companies that built the x86 architecture used in many applications today. PIC is known for their microcontrollers that are heavily used in industries. Arduino is an open source board that includes microcontrollers and can be used for a variety of applications. These technologies highlight the importance of microcontrollers and microprocessors today and how it advances the industry. As evident by this report, as the times progress, the technology and applications of these technologies will also increase.

Video Link:

<https://www.youtube.com/watch?v=NZY1TPlh8g4>

References:

www.arduino.cc

www.techopedia.com

www.electronicmedia.info

<https://www.techopedia.com/definition/5334/x86-architecture>

<https://www.digitalindustry.com/2020/01/07/intel-processor-shortage-how-it-can-affect-you/>

<https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/ia-introduction-basics-paper.pdf>

<https://www.elprocus.com/introduction-to-pic-microcontrollers-and-its-architecture/>

<https://www.watelectronics.com/pic-microcontroller-architecture-and-applications/>

<https://www.circuitstoday.com/peripheral-interface-controller-pic>

<https://electronicsdesk.com/pic-microcontroller.html>

BELOW AVERAGE

ELECTRICAL ENGINEERING DEPARTMENT



**Microprocessor Design
ELEG 3073-P01
Fall 2020
Dr. Cui**

14

PROJECT VON NEUMANN VS HARVARD

**Submitted By:
Christian Rodriguez**

**Date of Submission: 9/28/2020
Date of Performance: 10/12/2020**

Introduction:

Von Neumann and Harvard architectures are classified by how they use their memory function. Both architectures save data from microcontrollers; however, both architectures work differently in the way they store the data, which will be explained in this research paper with significant differences compared. Modern day computers have both architectures inside of them when saving data from computer functions performed by the user.

Theory:

Von Neumann architecture was developed by John von Neumann the purpose of his study was to find a way for a computer to store data compiled by the user. Von Neumann architecture uses different sets of data and address buses to read and write instructions to the computer. The first program Von Neumann program was able to store 96-word memory. "A true Von Neumann computer performs the following sequence of steps to execute a single instruction" check the memory address in the control unit's program counter." "Fetch the instruction located at the given address." "Add the length of instruction to the program counter. Execute the instruction using the control unit, which commands the rest of the computer to perform the instruction/operation. The instruction may change the address in the program counter for two reasons: If some arithmetic condition is true to perform a repetitive operation Proceed to step 1 and repeat the process for the next instruction". (http://wiki.sjs.org/wiki/index.php/History_of_Computers_-_Von_Neumann_Architecture)

A potential drawback of Von Neumann architecture is it can only perform one function at a time, which cases the processor to slow down resulting a long downloading speed. Von Neumann has potential computer crashes because of only performing one function at a time.

Harvard architecture works with separate pathways within the computer to gather instructions and run data. Harvard architecture uses a centralized unit and counters that storage as data. The program needs to be loaded by an operator; however, the processor cannot run its own function. While using Harvard architecture simple operations structure can differ throughout the system. Programmed task in Harvard architecture can be stored in read only memory while data requires read- write memory. (https://en.wikipedia.org/wiki/Harvard_architecture)

Contrast of Von Neumann and Harvard architecture:

Von Neumann architecture is based on the original functions of ancient computers that stored inside of the computer. Harvard architecture based on modern function of computers using relays inside of the computer. Von Neumann uses physical memory used for both instructions and retrieving data from the computer. Harvard architecture uses separate memory address for instructions and retrieving data from the computer. Von Neumann common bus for data and instruction transfer, while Harvard requires separate buses used for transferring data and instruction. In Von Neumann two clock cycles are required for a single function, while Harvard instruction is executed in a single cycle. Von Neumann is much cheaper to construct than Harvard architecture.

Discussion of Methodology:

Both Von Neumann and Harvard architecture had a big influence on the advancement of microprocessors in computers by being able to save data. Both architectures work in different

way, however both imply the same concept of saving data used by the computer user. The most common microprocessor used in modern day electronics is PIC16F it's very small in size but provides an very efficient amount of storage and power using both Von Neumann and Harvard architectures. "The basic physical organization of a modern computer, based on the *von Neumann architecture* model, comprises 5 units, namely memory, control, arithmetic-&-logic, input, and output. The CPU) comprises control and arithmetic-&-logic units."

(<https://www.sciencedirect.com/topics/computer-science/von-neumann-architecture#:~:text=The%20Von%20Neumann%20architecture%20consists,fetching%20and%20execution%20cycles%20seriously.>.)

These functions present in our modern-day microprocessors help us perform the essential task of saving a word document. These architectures helped microprocessors become smarter in using standard procedures, using sequential logic to register simple clock function tasks to save items used to either RAM or the CPU.

Discussion of Influence in Microprocessor Industry:

Von Neumann was first described by mathematician John Von Neumann in 1945 now 75 years ago, and much of his idea is still seen in modern day microprocessors. The basis of his idea are still seen today, how modern computers save simple functions from memory unit to house stored data. "A fundamental power of the von Neumann design is the concept of an instruction set, or a set of machine instructions strung together to create the program that will be executed. The ability to store either program instructions, or data, into the memory unit allows for arbitrary programs to be loaded into memory and executed — a feature that architectures prior to von Neumann (known as fixed-program computers) did not allow. Initially, this ability to arbitrarily manipulate instructions and data enabled the creation of tools like linkers, assemblers, and compilers."

(<https://info.dovermicrosystems.com/blog/von-neumann-cybersecurity>) This function allows our computers to provide the task needed to save and store information required to work properly and allow the microprocessor to run more efficiently and faster. Harvard architecture is seen in day to day life in Arduino Platform from using basic Arduino Uno board to complex functions we see Harvard architecture present. The Arduino board is based on the Harvard architecture where the program code and program data uses sperate memory storage. "This DSP utilizes a modified Harvard architecture consisting of separate program and data buses and separate memory spaces for program, data and I/O. It is an accumulator-based architecture. The basic building blocks of this DSP include program memory, data memory, ALU and shifters, multipliers, memory mapped registers, peripherals and a controller."

(<https://www.sciencedirect.com/topics/engineering/harvard-architecture>) A microprocessor will either have a Von Neumann or Harvard architecture not both in the same microprocessor. Harvard architecture is better than Von Neumann when talking about speed comparison with a faster processor.

Conclusion:

In conclusion, when comparing Von Neumann and Harvard architecture, the Harvard architecture is a better memory source for the microprocessor of choice because of its speed. However, learning about both architectures, we see a blueprint of Von Neumann, the original architecture of storing data made 75 years ago in modern day microprocessors. The first program Von Neumann program was able to keep 96-word memory. "A dedicated Von Neumann

computer performs the following sequence of steps to execute a single instruction. Technology has advanced dramatically since the start of the architectures in multiple boards and programs; we see both architectures. An example of this is when looking at the Arduino Uno, it uses a simple Harvard architecture. “The basic physical organization of a modern computer, based on the *von Neumann architecture* model.” (<https://www.sciencedirect.com/topics/computer-science/von-neumann-architecture#:~:text=The%20Von%20Neumann%20architecture%20consists,fetching%20and%20execution%20cycles%20seriously.>) We see advancement of technology in microprocessor design many of the aspects to run these memory functions from both architectures Von Neumann and Harvard.

Video Link Below:

<https://www.youtube.com/watch?v=F-vxKrwLIco>

Reference Page

<https://www.sciencedirect.com/topics/computer-science/von-neumann-architecture#:~:text=The%20Von%20Neumann%20architecture%20consists,fetching%20and%20execution%20cycles%20seriously.>

https://en.wikipedia.org/wiki/Harvard_architecture

http://wiki.sjs.org/wiki/index.php/History_of_Computers_-_Von_Neumann_Architecture

<https://info.dovermicrosystems.com/blog/von-neumann-cybersecurity>

<https://www.sciencedirect.com/topics/computer-science/von-neumann-architecture#:~:text=The%20Von%20Neumann%20architecture%20consists,fetching%20and%20execution%20cycles%20seriously.>

**MICROPROCESSOR SYSTEMS DESIGN
FALL 2020**

FINAL PROJECT

TOPICS COVERED IN FINAL PROJECT

FINAL PROJECT

**DESIGN MICROCONTROLLER BASED EMBEDDED SYSTEM TO
SOLVE REAL LIFE PROBLEM**

**ELECTRICAL AND COMPUTER ENGINEERING
DEPARTMENT
OUTCOMES SPECIFIC ASSIGNMENT COVER SHEET
ELEG 3073 MICROPROCESSOR SYSTEMS DESIGN
FALL 2020**

Instructor: Dr. Suxia Cui

Title of Assignment

Final Project

Outcome Measured Using this Assignment

Outcome 1

An ability to identify, formulate, and solve **complex engineering problems** by applying principles of engineering, science, and mathematic.

Check the engineering design characteristics related to this assignment

Assignment problem	
	involves wide-ranging or conflicting technical issues,
	has no obvious solution
	Addresses problems not encompassed by current standards and codes
	involves diverse groups of stakeholders
X	includes many component parts or sub-problems
	involves multiple disciplines
	has significant consequences in a range of contexts
Assignment problem involves the application of the principles of	
X	engineering
	science
	mathematics

Brief justification of the suitability of assignment for the outcome

Student will design a microcontroller-based system to be able to solve a real-life problem.

PRAIRIE VIEW A&M UNIVERSITY
COLLEGE OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
ELEG 3073 Microprocessors Systems Design
Project 2

In this mandatory project, students can work as a group of maximum 3 students toward a microcontroller/microprocessor related system design. You can choose one microcontroller, such as PIC, ARM, AVR, Intel, Arduino, or etc. The system must be able to meet one real life application need, and involve both hardware and software consideration. The project is due on Monday of week 15, and by week 12, each group needs to send instructor the names of your group members and the tentative project title. Each group needs to turn in one group written report of at least three pages (not including cover page) long with single space and 12 font. The following format should be observed for preparation of your written reports.

1. Cover Page: indicating title and all partners
2. Real life problem to be solved
3. Overview/background of this microcontroller/microprocessor
4. System design and implementation
5. Results and analysis (tables, graphs, figures)
6. Discussion/Explanation of how the applications influence microcontroller/microprocessor industry
7. Conclusion
8. References

This assignment is to measure ABET outcome [1]: An ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics

Each group is also required to provide a demonstration video to show the completion of the project. The video should be less than 8 minutes including a PowerPoint about the design (both hardware and software).

Your project will be evaluated as a group with the rubric:

Quality of Visuals and or written materials (i.e readable text, no typos, figures) (1.1)	_____ (6 points)
Demonstrated knowledge of research and quality of the designed system (1.2)	_____ (8 points)
Delivery of Information (quality of presentation & demo, organization, clarity) (1.1)	_____ (4 points)
Ability to address questions, anticipated issues posed during presentation. (1.1)	_____ (2 points)
TOTAL	_____ (20 points)

ABOVE AVERAGE

SAMPLE GRADING SCHEME AND GRADED FINAL REPORT FROM STUDENT GROUP

GRADING SCHEME FOR DESIGN CONTENT

	Points Assigned	Points Received	% for PC
1. Demonstrated knowledge of research and quality of the designed system (1.2)	8	7	89
2. Quality of Visuals and or written materials (i.e readable text, no typos, figures) (1.1)	6	5.5	91
3. Delivery of Information (quality of presentation & demo, organization, clarity) (1.1)	4	4	
4. Ability to address questions, anticipated issues posed during presentation. (1.1)	2	1.5	
TOTAL	20	18	90

ELECTRICAL ENGINEERING DEPARTMENT
PRAIRIE VIEW A&M UNIVERSITY



Microprocessor Project 1

ELEG-3073

Fall 2020

Group Names:

George Polk

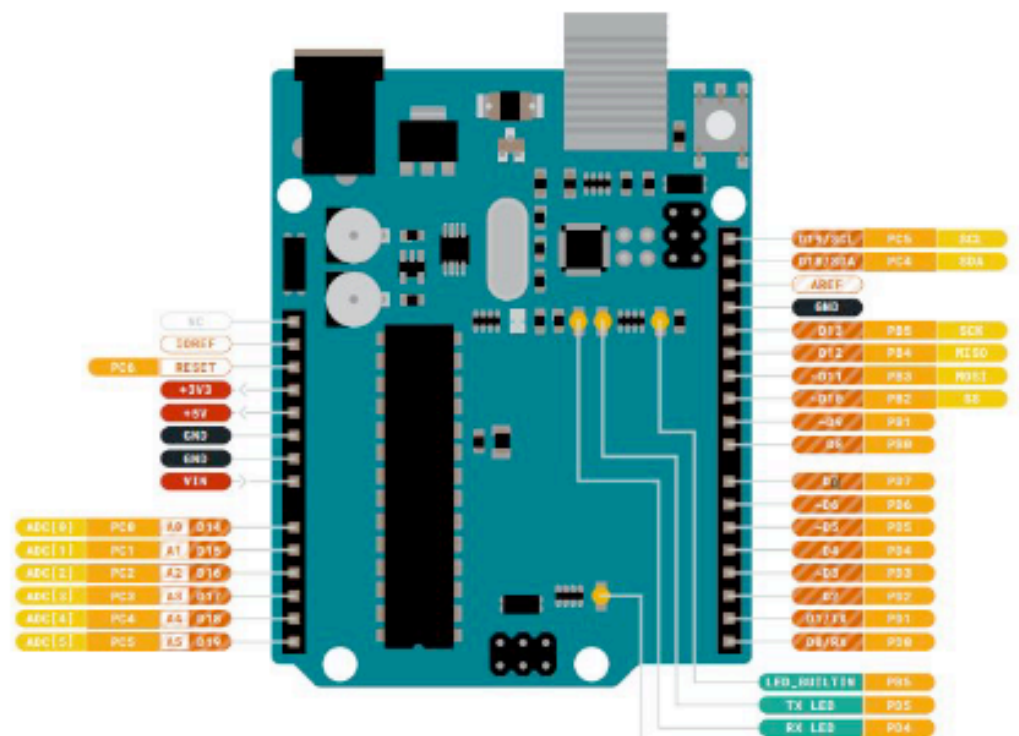
Hiram Pouncy

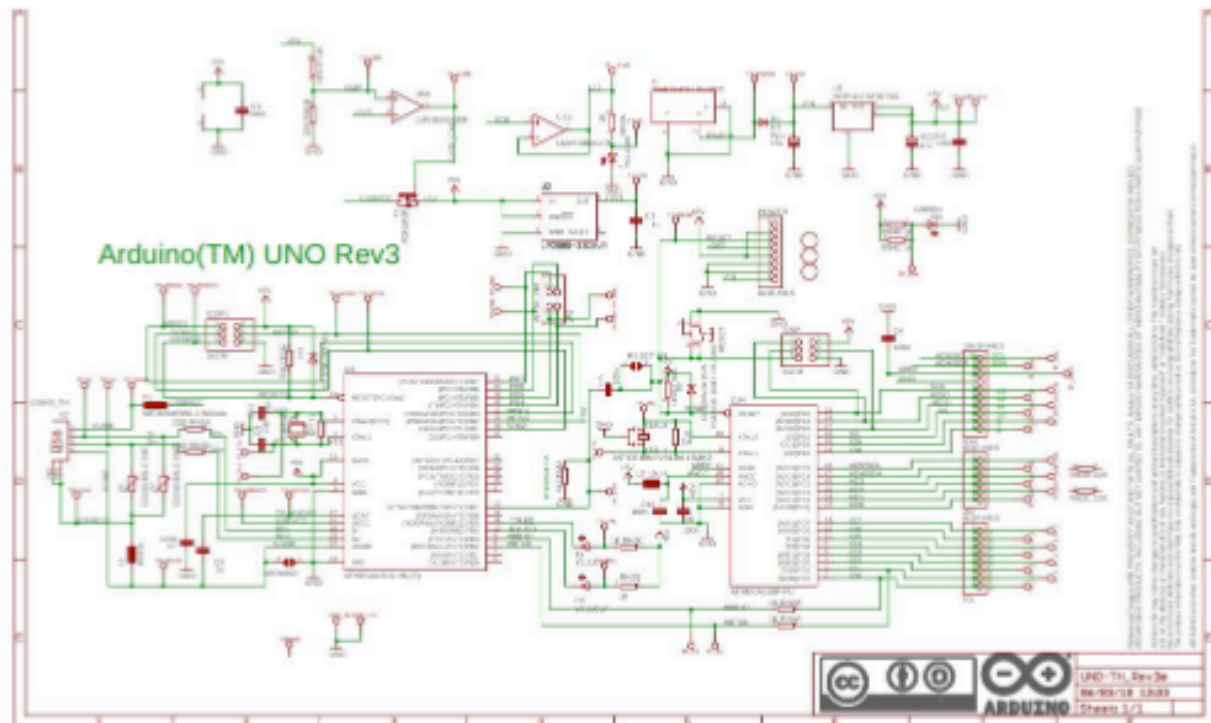
Matthew Barnett

Overview/Background

In 2005 the first Arduino board was created in Ivera, Italy. The idea of the Arduino board was conceived in the Interactive Design Institute in Ivera, Italy by a masters student. The name of the master student is Hernando Barragan, Hernando came up with the idea while writing his thesis called "Arduino-La rivoluzione dell'open hardware". The work done by Hernando Barraga was eventually used by one of Arduino's co founders Massimo Banzi, Massimo Banzi was also the professor of Hernando Barragan. The overall goal of the Arduino is to give users of electronics an inexpensive platform to prototype and build electronics. The Arduino would also make building different types of electronics easier.

The tem working on the Arduino received most of their ideas from Hernando Barragan's Wiring project. The Wiring prototyping platform has an easy to use IDE and a ready to use circuit board. The Wiring project became a helpful tool in the success of Arduino.





There are different versions of the Arduino boards currently available, some of the different versions include: Uno Rev3, Leonardo, Nano, and Micro. The Arduino Uno board, which is one of their most popular boards, uses an ATmega328P microcontroller. The ATmega328P has 14 digital input/output pins, 6 analog inputs, a USB connection, a 16 MHz ceramic resonator, an ICSP header, and a reset button. The Arduino Uno board, as shown above, has all of the required hardware in order for the microcontroller to be fully functional. To power the Arduino Uno board either connect it to a computer with a USB cable or power the board with a AC to DC adapter.

Real Life Problem to be Solved

A problem that is a huge issue in today's society is security. Keeping valuable items and information secure is a monumental as well as important task. There are many ways to make sure that valuable information and items are secure, one way being using a RFID reader.

To design a RFID reader there are several components that are needed. The hardware components that needed are an Arduino Uno board, A SparkFun RFID Reader Breakout, and an Adafruit RFID/NFC card. The software that is needed to program the Arduino board is the open-source Arduino software called Arduino IDE 1.8.13.

The RFID reader works by sending electromagnetic waves to send data over small distances. In a RFID system there is a tag that is attached to the object to be identified by the reader.

Code

```
#include <SPI.h>

#include <MFRC522.h>

#define SS_PIN 10
#define RST_PIN 9
MFRC522 mfc522(SS_PIN, RST_PIN); // Create MFRC522 instance.

void setup()
{
  Serial.begin(9600); // Initiate a serial communication
  SPI.begin(); // Initiate SPI bus
  mfc522.PCD_Init(); // Initiate MFRC522
  Serial.println("Approximate your card to the reader...");
  Serial.println();
}

void loop()
{
  // Look for new cards
  if ( ! mfc522.PICC_IsNewCardPresent())
  {
    return;
  }
}
```

```

}
// Select one of the cards
if ( ! mfrc522.PICC_ReadCardSerial())
{
    return;
}
//Show UID on serial monitor
Serial.print("UID tag :");
String content= "";
byte letter;
for (byte i = 0; i < mfrc522.uid.size; i++)
{
    Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
    Serial.print(mfrc522.uid.uidByte[i], HEX);
    content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
    content.concat(String(mfrc522.uid.uidByte[i], HEX));
}
Serial.println();
Serial.print("Message : ");
content.toUpperCase();

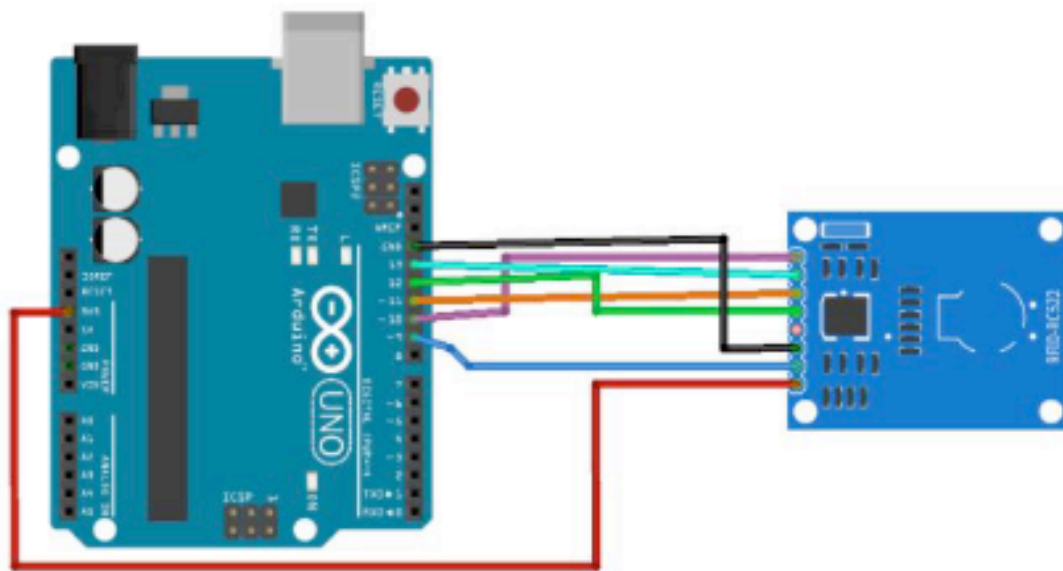
if (content.substring(1) == "BD 31 15 2B") //change here the UID of the card/cards that
you want to give access
{
    Serial.println("Authorized access");
    Serial.println();
    delay(3000);
}

```

```

else {
  Serial.println(" Access denied");
  delay(3000);
}
}

```



Schematic 1

Influence of the Arduino Board

The Arduino Board has made such a huge impact on the microcontroller/processing industry for the simple fact that they are so easily accessible and simple to master. They are used in almost every classroom to teach people the basics of microcontrollers. What makes this such an impactful product is that it influenced companies to not so much focus on controllers and processors that can be used for industrial purposes but for recreational purposes as well. James Provost, a routine reporter of everything Arduino

states that "The Arduino is simple, but not too simple. It's built around the idea that students will be using these to "do" something: get sensor data in, have a bit of code, do something with that." He states in his articles that the arduino was so well accepted and used because it was simple and effective for students due to its interchangeability and low cost. He also talks about its open source code, sensors, and how its not made by everyday chip makers. There are many controllers that brought about revolutionary changes from an industrial viewpoint in regards to the processors/controller industry. The effect the arduino had however was that it opened the door for students to be able to access microcontrollers/processors without starting with something too complicated. Provost talks about how there is a new "Arduino Killer" brought about with the hopes of taking the crown from the current king but all these products have is hope and no solid evidence.

Conclusion

Within this report we covered the history of the Arduino Boards from the first one being built in 2005 up until today's use of it. One interesting fact was that the board was made by a master student and is now used by many other students. We also covered how it revolutionized the microcontroller/processor industry simply because it was one of if not the first board made to be used by college students and beginners looking to get into controllers and processors. We also included a real world problem in our society and how the Arduino board is incorporated in solving this problem. According to Provost there are roughly 150k and up boards on the market since 2011, he believes in time these boards will be used in every school. There are no boards that provide the simplicity, interchangeability, and low cost on the level that the Arduino does. The board caters to college students and teachers way better than other boards in the industry today which is why the board is truly the ruler of the classroom.

Video Presentation

<https://youtu.be/L9p4KVTRglo>

References:

<https://arduinohistory.github.io/>

<https://www.circuitstoday.com/story-and-history-of-development-of-arduino>

<https://store.arduino.cc/usa/arduino-uno-rev3>

<https://www.deviceplus.com/arduino/how-did-it-begin-the-history-of-arduino/>

<https://www.arduino.cc/en/Guide/Introduction>

[Security Access Using RFID Reader - Arduino Project Hub](#)

<https://makezine.com/2011/02/10/why-the-arduino-won-and-why-its-here-to-stay/>

AVERAGE

SAMPLE GRADING SCHEME AND GRADED FINAL REPORT FROM STUDENT GROUP

GRADING SCHEME FOR DESIGN CONTENT

	Points Assigned	Points Received	% for PC
1. Demonstrated knowledge of research and quality of the designed system (1.2)	8	7	89
2. Quality of Visuals and or written materials (i.e readable text, no typos, figures) (1.1)	6	5	83
3. Delivery of Information (quality of presentation & demo, organization, clarity) (1.1)	4	4	
4. Ability to address questions, anticipated issues posed during presentation. (1.1)	2	1	
TOTAL	20	17	85

Parking Management Project
ELEG 3073 Microprocessors Systems Design
Project 2



By: Marvin Berrios

Arduino Overview:

The Arduino project was started at the Interaction Design Institute in Ivrea, Italy. In 2003, Hernando Barragan created the development platform 'Wiring.' In 2005, Massimo Banzi, David Mellis, David Cuartielles, and two others extended 'Wiring' by adding support for the cheaper ATmega8 microcontroller, which was then called Arduino. This is where Arduino took off. In 2008, the five co-founders of the Arduino project created a company, Arduino LLC, to hold trademarks associated with Arduino.

Early Arduino boards came with an RS-232 serial interface and an Atmel ATmega8 microcontroller chip. It had 14 digital input/output pins and 6 analog input pins, and the power connector at the lower left. Newer boards are similar but have more features.

Arduino is an open-source electronics platform based on easy-to-use hardware and software, it is a development board which contains a microcontroller mainly 8-bit AVR. It is designed to make electronics more accessible to artists, designers, and anyone interested in creating interactive environments or objects. The Arduino IDE is the open-source software, the coding is easy to write and upload to the board. It runs on Windows, Linux, and Mac OS X. It can be used with any type of Arduino board.

Arduino products are very cheap compared to other more complex and expensive microcontroller-based boards. The hardware is easy to use and so is the software, Arduino IDE. Even though Arduino is very useful for various purposes, it cannot do too complex functions due to its limited hardware. Arduino is more of a beginner's guide into electronics and basic projects dealing with microcontrollers. For advanced projects Arduino would probably be useful in initial phases but not for completion stages. Arduino Products include but are not limited to: ATmega8, ATmega168, ATmega368, ATmega1280.

Real life problem to be solved:

All throughout the United States urban cities encounter the problem of finding a free parking spot in cities during the busy hours of the day. Most of the common areas that encounter this problem include public parking, universities, schools, stadiums, and shopping areas. With limited parking spaces, finding an available space in traffic filled parking lot can be time consuming and frustrating, especially when an individual is already running late.

The solution to this problem is the Parking Management project I worked on. Through the use of technology parking management can be made more efficient. The project goal is to reduce traffic and provide information on parking lot availabilities to individuals ahead of time. The rapid increase of population in urban areas also increases the number of vehicles in the area. The more vehicles the more parking that is needed in public areas. The Parking Management project can resolve this in an efficient manner.

System design and implementation:

In terms of hardware, the Parking Management project is based off an Arduino Mega 2560 board that is connected to a breadboard that has three HS 2604 Ultrasonic sensors. (Seen below in Figure 1) The software this project uses includes the Arduino Integrated Development Environment (IDE) application, which uses the C++ programming language. As for the user and admin clients, they are developed using the Visual Studio application and through this the programming languages used were C#, JavaScript and CSS. (Visual Studio interface seen below in Figure 2) This information was transmitted to the cloud server Amazon Web Services, where data transmission, collection, and analysis were done and used to provide parking space availability and other parking information.

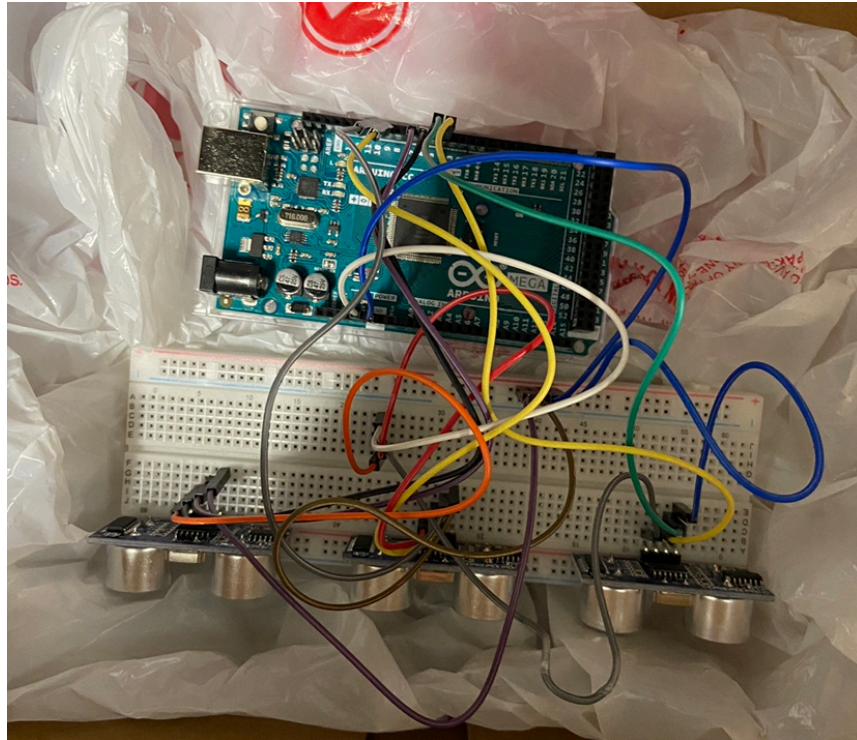


Figure 1: Arduino Board & Ultrasonic Sensors

The way it works is the Arduino board captures the readings of each of the Ultrasonic sensors based on the programming done in the Arduino IDE application and then it displays either “Occupied” or “Available” for each sensor. Each of the three Ultrasonic sensors is considered a parking spot in this project. From here the coding done in the Visual Studio application displays an interface which shows a display in a much cleaner look. Amazon Web Services then does the data compression and processing and finally it becomes an application for facilitating parking in clustered urban areas.

An example of the Admin client display is shown below. This is where the Admins have access to the display of the Parking Management project. There is also a user interface which has less options because it is the display that the user would see as they are looking for a parking spot. From hardware to software, the combination of both creates an innovative way of managing parking spots and parking lots wherever desired.

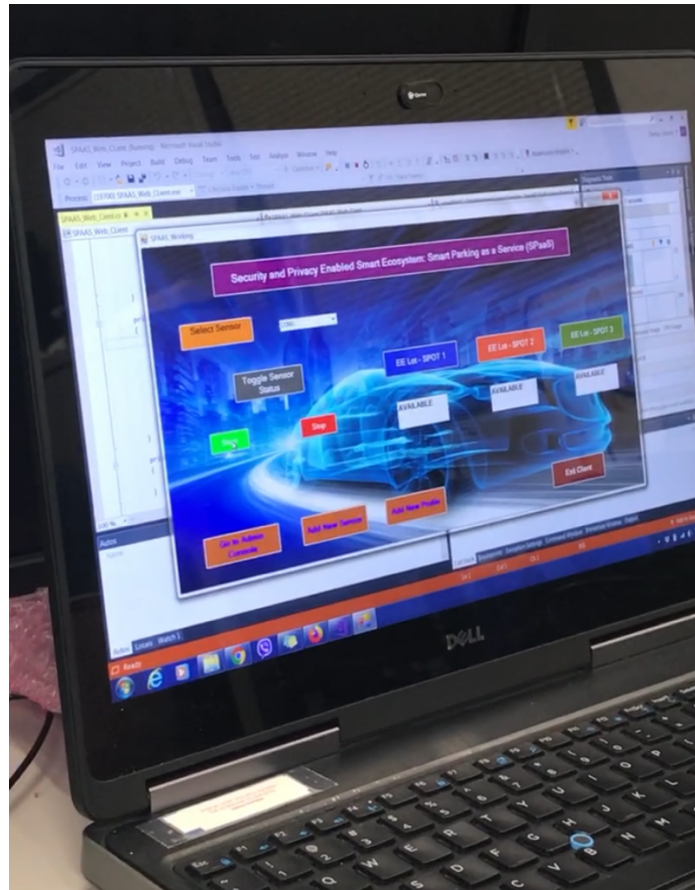


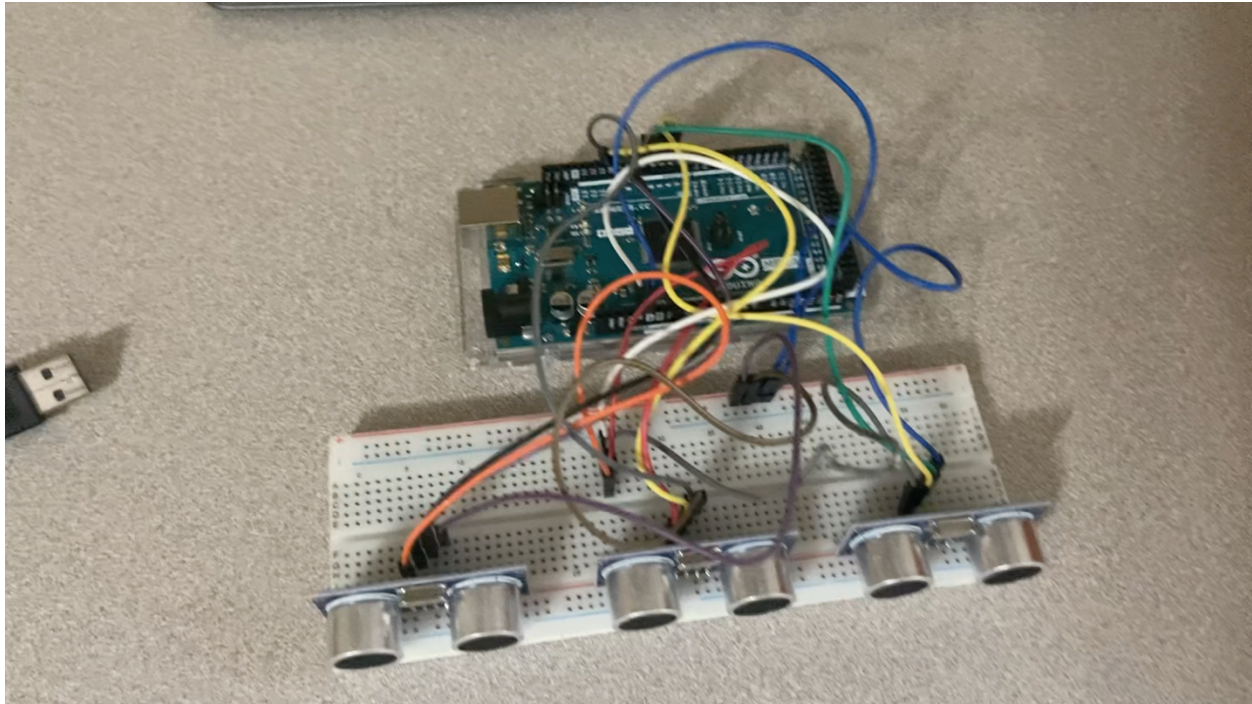
Figure 2: Visual Studios Admin Interface

Discussion & Conclusion:

Arduino products have helped with the development of microcontrollers and the importance of their use since 2003. This platform can be used by people of all ages and all levels of experience. This is what makes Arduino so important in terms of applications in the microcontroller industry.

The overall goal of this project outcomes the efficient use and the various ways of alleviating the common problem of parking spaces in big cities during busy times. The sensor network along with the software technologies create a system that could be innovated and one day be used in public urban cities. The system is user-friendly, therefore making it easy for anyone to use in their daily lives. Microprocessors and microcontrollers contribute to many of the innovations that change the world. Slow progress is still progress, and with the use of such microcontrollers and microprocessors new technologies can emerge and make change around the world.

Video:



References:

<https://www.arduino.cc>
<https://flaviocopes.com/arduino-programming-language/>
<https://aws.amazon.com>
<https://mobidev.biz/blog/iot-based-smart-parking-system>
<https://ieeexplore.ieee.org/document/6569312>

BELOW AVERAGE

SAMPLE GRADING SCHEME AND GRADED FINAL REPORT FROM STUDENT GROUP

GRADING SCHEME FOR DESIGN CONTENT

	Points Assigned	Points Received	% for PC
1. Demonstrated knowledge of research and quality of the designed system (1.2)	8	6	75
2. Quality of Visuals and or written materials (i.e readable text, no typos, figures) (1.1)	6	4	75
3. Delivery of Information (quality of presentation & demo, organization, clarity) (1.1)	4	4	
4. Ability to address questions, anticipated issues posed during presentation. (1.1)	2	1	
TOTAL	20	15	75

Microprocessor ELEG 3073
Final Project
History, Architecture, & Applications
of Intel



By: Javian Walker
Date: 11/27/2020

Introduction/Overview:

The history of Intel processors dates back from the early 1970s to today, starting with the 4004 chip. The 4004 chip was manufactured from 1971 to 1981. It became the first commercially available processor, and the very first complete CPU on a single chip. The chip was built and packaged in a 16-pin ceramic dual in-line package and was initially released on the market with a speed of 108 KHz while having 2,300 transistors and delivered a performance of 0.07 MIPS. This processing chip was groundbreaking, however it was replaced in 1972 by the 8-bit 8008 chip. This chip was released on the market with a 0.5-0.8 MHz clock speed and 3,500 transistors.

Then shortly in 1974, the 8080 chip was manufactured with 4,500 transistors and a clock speed up to 2 MHz. The 8080 then went on to become famous for being used in the Altair 8800 microcomputer, and Boeing's AGM-86 cruise missile. Even with all this short and quick success, none of these chips were sold in considerable volumes, but that did not decrease the demand for such microprocessing chips. With that being the case, microprocessing chips then begin to evolve and are mass produced. By the early 2000s, Intel had built chips such as the Pentium 4 and Xeon, that were integrated into everyday technology. These chips then progress over the next two decades and evolve into what we know as the Core i-series - Haswell, chips that have 18 cores with the Haswell-EP line of Xeon processors, with up to 5.69 billion transistors and clock speeds of up to 4.4 GHz and are used in just about every device that has circuit board in it.

Theory & Applications:

The Intel Core chips are pipelined based in architecture. Meaning, instructions move through a number of internal stages, from entering the processor to being completed. As an instruction is completed and exits a stage, another instruction can enter. This is in return minimising the idle time for each internal component. Each core chip has an estimation of at least fourteen stages in its pipeline, but there are a number of complications within the chips, such as early completion or out of order execution, it is difficult to determine exactly how many stages there are in the pipeline.

Now based on this knowledge, Intel chips are very successful inside of computers and other smart devices, due to handling its various commands and applications. People typically would like for their devices to move fast, respond quickly, and handle multiple things at one time. With the pipelined architecture, the processors don't have to sit and hold on to instructions. This is what allows for a user of such a laptop or smart device to be able to open multiple windows, and complete multiple tasks all at the same time without slowing the device down to a point and making it so slow to where it is unresponsive. And due to this fact, this is why there is constant development and new applications of microprocessors so they can be more improved, faster, and be able to handle larger tasks in short periods of time.

Discussion of Methodology:

Intel's developed methodology is based on this ideal of process-architecture-optimization. Which is essentially their, three step or three main focal points when developing microprocessing chips. In the Process stage, Intel's focus is to develop and produce a slightly enhanced processor core, along with a new graphics processor. Which is also manufactured on a significantly denser materials or technology, than the product materials that were manufactured before it. This is a direct response and or replacement to the "tick" ideal in Intel's previous methodology.

In the Architecture stage, Intel will typically introduce a new processor core with significantly improved performance and feature enhancements, which in theory ranks it over the previous generation. Also, Intel would introduce a new graphics processor in the architecture step, which of course would have significant platform feature enhancements from the previous generation. And following the logic from the Process stage, all this will be built on the same advanced materials or technology that is being used in the Process stage. Keeping the materials the same throughout the entire methodology would be a direct response and or replacement for the "tock" ideal in the previous methodology..

Finally in the Optimization stage, Intel would essentially have the same architecture throughout the entire process, and introduce new implements of it on a performance-enhanced version of the manufacturing materials or technology that was used in the previous stage. Meaning, they are looking at past generations of processors, improving each aspect of the processor even the materials it is being built on, while keep the overall structure of the processor the same. And with this methodology, Intel can develop and produce new and better products each year effortlessly which is a clear positive for them as a company and as long as they are willing to put in the work and meet new challenges as they appear, there will always be room for improvement with their processors.

Discussion of Industry:

Based on the constant improvements and development of new applications, the market in the industry of course grew significantly. This led to the manufacturing of other brands and types of processing chips all of which were built differently and designed to handle different tasks from other processing chips. Regardless, the overall core of the chips function remained the same despite its physical design. All at the same, the chips needed to be cost efficient. Meaning, there is a high demand in the industry for processor chips because they are used in just about every smart device but they have to be cheap enough for mass production. Even though not all of the

chips are cheap, some are fairly expensive with a price range of \$42 all the way up to \$2000, this is just depending on the system the chips are being produced for.

Real life problem to be solved:

One real life problem that can be solved using intel processing chips is the ability to detect cyberbullying. There have been early developments for this technology using the intel processing chips with a combination of AI (artificial intelligence) earlier this year by a few young kids in India. The model that they developed has some flaws but even still it was able to detect up to 80% of cyberbullying with accuracy. Even with the advancements, cyberbullying is still at an all time high due to the exponential growth of social media in everyday use among adolescents.

System design and implementation:

In today's society, the hardware used for AI today mainly consists of one or more of the following designs for microprocessors CPU (Central Processing Units), GPU (Graphics Processing Units), FPGA (Field Programmable Gate Arrays), or ASIC (Application Specific Integrated Circuits). With these designs, the integration of microprocessors and AI becomes a lot more easier and efficient. We need the AI to be cutting edge and efficient because we need the AI to be able to handle and process large amounts of algorithms. These algorithms are used to recognize things like videos & images, sound & speech, text analysis etc. And at this point is where the software comes into play, because the chips have to be designed and programmed to process these algorithms.

Results and Analysis:

*See powerpoint for the results & analysis (tables, graphs, figures)

Explanation of how the Applications influenced Industry:

As stated earlier in my report, the microprocessors, especially the intel chips, have a heavy influence on the industry. Some studies have shown that the demand for AI chips and its application-specific hardware will increase by about 10 to 15 percent. This could result in a \$109 billion dollar increase in the AI market by 2025. That is just 5 years from now, and technology is steadily growing and increasing at a rapid rate. Just imagine the influence these chips could have

in the next 10, 15 or possibly 20 years especially when the demand grows for further applications.

Conclusion:

In conclusion, Intel is one of the many processor chips that are mass produced with quality performance and essentially available to consumers year round. They have been around since the early 1970s and have grown and improved each year since its first arrival with the 4004 chip. Today, Intel's processor chips are used globally and in just about any device you can think of, especially laptops and computers. Also, in recent research, you can see many different applications for Intel chips. One of these applications is in AI and artificial intelligence is used worldwide. As long as there are circuit boards and advanced technology, there will be the demand for processor chips.

References:

- <https://www.businessnewsdaily.com/10817-slideshow-intel-processors-over-the-years.html>
- <https://www.zdnet.com/article/inside-intels-core-architecture/>
- <https://www.fool.com/investing/2016/09/18/intel-corporation-needs-to-tweak-its-product-devel.aspx>
- [https://www.cpu-world.com/Price_Compare/Desktop_CPU_prices_\(latest\).html](https://www.cpu-world.com/Price_Compare/Desktop_CPU_prices_(latest).html)
- [AI for Youth Uses Intel Technology to Solve Real-World Problems | Intel Newsroom](#)
- [AI hardware - What they are and why they matter in 2020 \(roboticsbiz.com\)](#)