

Architecture of a General-Purpose Processor

Computer Architecture Exploitation and Security

Table of Contents

[Objectives 2](#_Toc40177493)

[Background Reading 2](#_Toc40177494)

[Important Information 2](#_Toc40177495)

[Introduction 3](#_Toc40177496)

[Problem 1 5](#_Toc40177497)

[Problem 2 5](#_Toc40177498)

[Problem 3 5](#_Toc40177499)

[Problem 4 6](#_Toc40177500)

[Problem 5 6](#_Toc40177501)

[Problem 6 6](#_Toc40177502)

Problem 7………………………………………………………………………………………………….7

L*abs must be submitted by the due date for full credit. After due date late submissions will be accepted for a period of one week (seven days) and the grade will be reduced by ten percent (10%) per day after due day.* ***Assignments that are submitted more than seven days late will receive a grade of zero (0).***

I certify that the work submitted in this assignment is my own and that it has not been taken in whole or in part from any other source. I understand that the penalty for plagiarism will include a grade of zero (0) for this assignment plus disciplinary action in accordance with SAIT policies.

Signature: \_\_\_\_Coleton Sanheim\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# EVALUATION:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | Number Conversions | 8 |  |
| 2 | Negative Numbers and 2’ complement | 3 |  |
| 3 | Logical Operations | 11 |  |
| 4 | Signed/Unsigned Numbers | 7 |  |
| 5 | Von Neumann architecture | 4 |  |
| 6 | Harvard Architecture | 5 |  |
|  | TOTAL MARK | 37 |  |

Computer Architecture Exploitation and Security

Lab One

Objectives

This lab focuses on the following objectives:

* Review number systems.
* Describe the building blocks of a processor.
* Define Harvard and von Neumann architectures.
* Discuss the advantages and disadvantages of Harvard and von Neumann architectures.
* Define RISC and CISC architectures.
* Discuss the advantages and disadvantages of RISC and CISC architectures.
* Explain the interaction among the building blocks.
* Describe byte, word, doubleword and quadword.
* Perform processor-activity simulation (by hand).

Background Reading

See Instructor for any readings you may require to successfully complete this lab.

# Important Information

**DO NOT SUBMIT YOUR ANSWERS IN THIS DOCUMENT. CREATE A BLANK DOCUMENT AND SUBMIT YOUR ANSWERS THERE.**

**YOU WILL LOSE MARKS FOR NOT FOLLOWING THE ABOVE REQUIREMENTS.**

# Introduction

For this lab you can use any decimal calculator. **Calculators capable of hexadecimal or any base-2 operations are not allowed. Questions to appear on quizzes.**

Powers of 2

|  |  |  |
| --- | --- | --- |
| 0 | 1 | 0000 0000 0000 0001 |
| 1 | 2 | 0000 0000 0000 0010 |
| 2 | 4 | 0000 0000 0000 0100 |
| 3 | 8 | 0000 0000 0000 1000 |
| 4 | 16 | 0000 0000 0001 0000 |
| 5 | 32 | 0000 0000 0010 0000 |
| 6 | 64 | 0000 0000 0100 0000 |
| 7 | 128 | 0000 0000 1000 0000 |
| 8 | 256 | 0000 0001 0000 0000 |
| 9 | 512 | 0000 0010 0000 0000 |
| 10 | 1024 | 0000 0100 0000 0000 |
| 11 | 2048 | 0000 1000 0000 0000 |
| 12 | 4096 | 0001 0000 0000 0000 |
| 13 | 8192 | 0010 0000 0000 0000 |
| 14 | 16384 | 0100 0000 0000 0000 |
| 15 | 32768 | 1000 0000 0000 0000 |
| 16 | 65536 | 1 0000 0000 0000 0000 |

# Problem 1- Number Conversions \_\_/8

To easily identify the number format being used, the following rules are typically applied:

1. Decimal numbers are written in the ordinary form eg ***100, 200, 1 etc***
2. Binary numbers are prefixed by "0b" eg ***0b00001010, 0b11000011***
3. Hexadecimal numbers are prefixed by "0x" eg ***0x0A, 0xC3***
   1. Hexadecimal numbers can also be post fixed with the letter "h"
4. Octal numbers have a leading zero eg: ***0712, 0432***

Perform the following number conversions, make sure you represent the numbers properly – (**0x12 is not the same as 12**):

|  |  |  |  |
| --- | --- | --- | --- |
| **Number to Convert** | **Binary** | **Hexadecimal** | **Decimal** |
| 0x1400 | 0b1010000000000 | 0x1400 | 5120 |
| 64 | 0b1000000 | 0x40 | 64 |
| 0x248 | 0b1001001000 | 0x248 | 584 |
| 0b1101001101101100 | 0b1101001101101100 | 0xD36C | 54124 |
| 2304 | 0b100100000000 | 0x900 | 2304 |
| 0x1238 | 0b1001000111000 | 0x1238 | 4664 |
| 720 | 0b1011010000 | 0x2D0 | 720 |
| 0b11010100 | 0b11010100 | 0xD4 | 212 |

# Problem 2 Negative Numbers \_\_/3

Use 8-bit 2’s complement to find the negative value of the following numbers.

1. Show the procedure of each calculation.
2. **Answer should be in binary**.
3. -128

**10000000**

1. 0x85

**1111010**

1. Is it possible to have a signed 8-bit representation for the number 128?

***In other words, does 128 fall inside the range of a signed 8bit number.***

**No it is not possible, the range stops at 127**

# Problem 3 Logical Operations \_\_/11

|  |  |  |  |
| --- | --- | --- | --- |
| **Operand 1** | **Operand 2** | **Operation** | **Result** |
| 11 | 1 | ADD | **0xC** |
| 83 | 39 | ADD | **0x7A** |
| 6145 | 2056 | SUB | **0xFF9** |
| 60845 | 4680 | OR | **0xFFED** |
| 6055 | 4685 | XOR | **0x5EA** |
| 0xecb7 | 0xc0c0 | SUB | **0x2BF7** |
| 245 | 255 | AND | **0xF5** |
| 17 | 127 | OR | **0x7F** |
| 0xFF | 0xFF | XOR | **0x0** |
| 406 | 256 | AND | **0x100** |
| 300 |  | NOT **(~)** | **0x2D3** |

Perform the following number operations, the Result column should be in **hexadecimal** format. Do not use more digits that you need to. For example, if the result is 300, 16bits is sufficient, therefore don’t present the answer with more than 16bits

# Problem 4 Signed/Unsigned Numbers \_\_\_/7

1. Copy the following program to your VM and call it lab1\_4.c
2. Compile it: gcc lab11.c -o lab11

|  |
| --- |
| #include<stdio.h>  int main()  {      char lval = 0;      printf("=============================================================\n");      printf("    %50s    \n", "uint8 used as a shorthand for unsigned int 8bits");      printf(" %58s  \n","uint32 would be used as a shorthand for unsigned int 32bits");      printf("=============================================================\n");        printf("\n\tEnter a value in the range 0 -> 255 or -128 -> 127: ");      scanf(" %hhd", &lval);        printf("\n\t==========================================================");      printf("\n\tuint8 Hex | uint8 Char | int8 Hex | int8 Char | Binary\n");      printf("\t==========================================================\n");      printf("\t   %#02x %12d       %#02hhx %8hhd   ", (unsigned char)lval, (unsigned char)lval, (char)lval, (char)lval);      printf("%4s 0b", " ");      for(int count = 7; count > -1; count--)          printf("%d", (lval >> count) & 1);      printf("\n");  } |

1. Execute the program and enter the following numbers. Show the screenshot for each execution see the example execution below: **3pts**

|  |  |
| --- | --- |
| **Number** | **Execution** |
| **122** |  |
| **100** |  |
| **127** |  |
| **128** |  |
| **129** |  |
| **200** |  |
| **256** |  |

## Question

1. Why does the unsigned value for 256 result in the value of 0? **1pt**

**Its greater than the range of 255 that 8 bits can represent**

1. Why does the entered value of 128 result in the output values of -128 and 128? **1pt**

**The -128 is represented a signed number while the 128 is an unsigned number, and you can never see 128 in a signed 8 bit number as the range stops at 127**

1. Based on the input values; write a formula to predict the output of the unsigned values if the user entered values in the range 0-255? **2pts**

**if input = 0 -> 127, output = 0 -> 127 respectively**

**if input = 128 -> 255, output = -128 -> -1**

# Problem 5 von Neumann Architecture \_\_\_/4

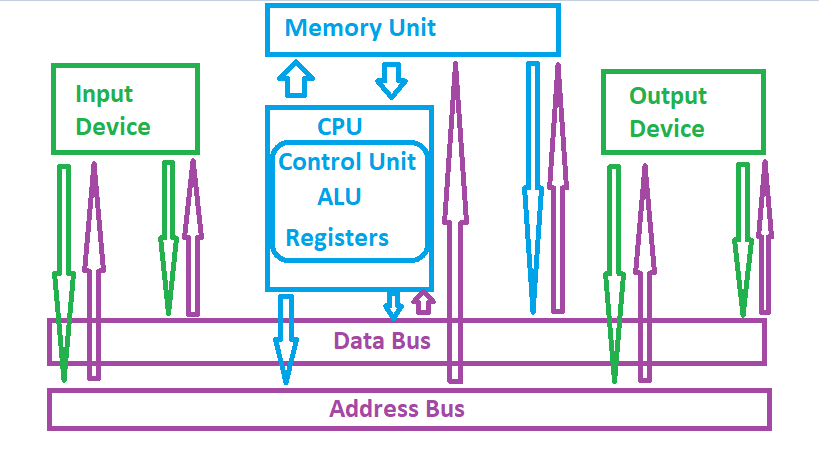
Draw a block diagram of von Neumann processor. You can use the shapes feature of Word to create your block diagram.

The diagram should have the following building blocks:

1. ALU
2. Registers
3. Data bus
4. Address bus
5. RAM

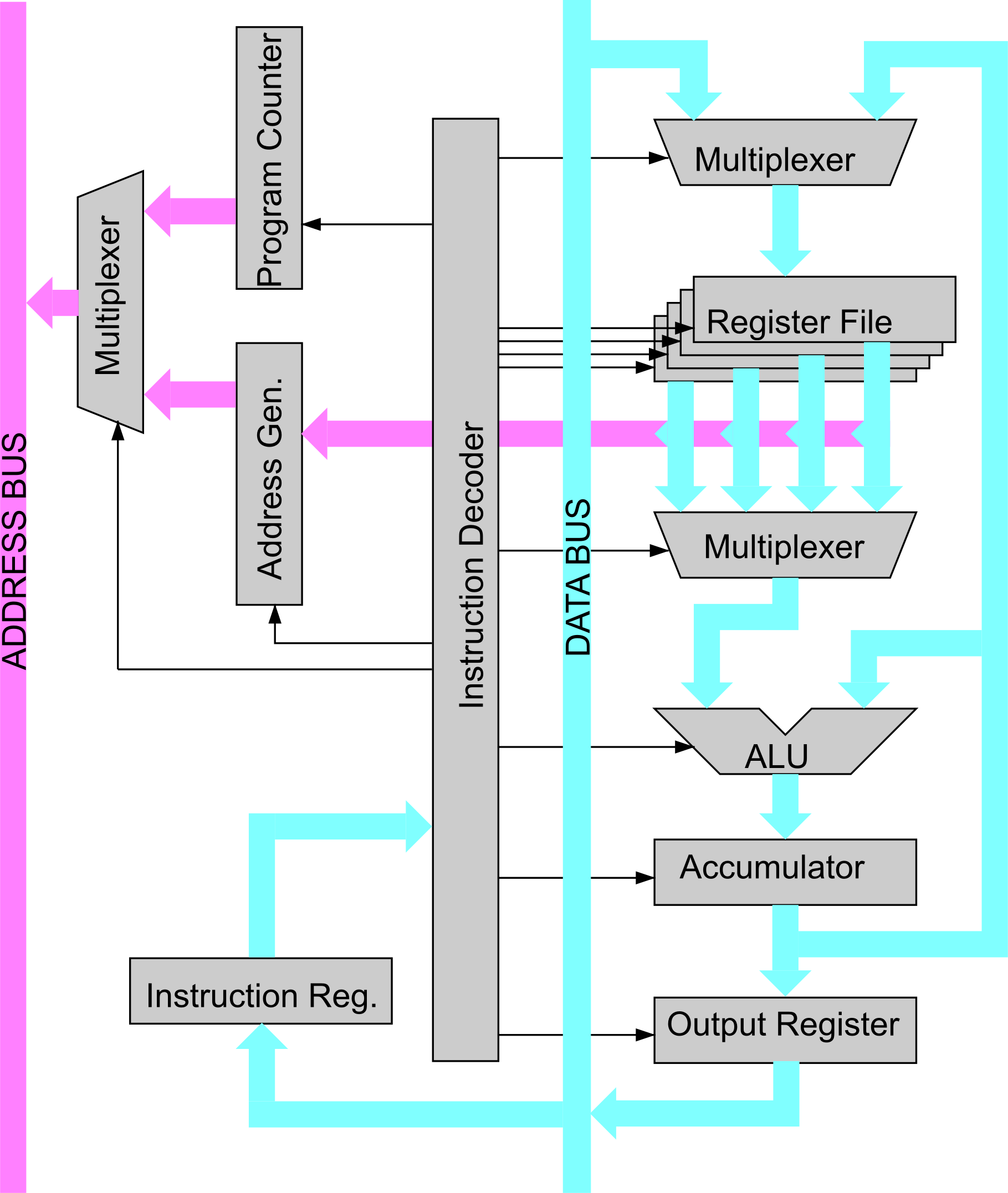
The diagram should show how each component is connected within the system. Some connections are bidirectional and others unidirectional and your diagram should reflect that. If you decide to create your diagram by hand make sure that the image is neat, use a ruler and make sure your handwriting is legible.

Attach the screen capture of the drawing

****

# Problem 6 Harvard architecture \_\_\_/5

Describe how the Generic CPU illustrated on the following page would have to be modified to conform to the Harvard architecture.



**You would have to alter the data bus so the instruction memory does not access it, other that that it should conform to Harvard architecture as far as I can tell**

© 2017, Southern Alberta Institute of Technology