

First Question

- a) The ALU is the arithmetic logic unit and it will be used to perform mathematical operations such as addition, subtraction, multiplication and division. It can also perform logic operations such as OR and AND

The program counter stores the address of the next instruction.

TRISA is a data direction register which determines whether each I/O pin is an input or output. Setting a TRISA bit (= 1) will make it an input. Clearing a TRISA bit (= 0) will make it an output.

- b) This is due to some ARM commands being longer and more complex than others.

- c) RISC means reduced instruction set computer. A RISC device will have fewer cycles per instruction.

Some of the advantages of a RISC device are:

- instructions take one cycle time
- Pipelining is easy
- More registers.

Some disadvantages are

- Complexity in compiler
- Require fast memory systems.

Connor Daniel Sheet 2

- d) For a time critical process I would use ~~asm~~ assembly language. This is due to it ^{being} faster in speed and allows complex jobs to run in a simpler way.

Question 2.

- a) The main purpose of the Stack is to store the program counter. An example of it's would use would be when a subroutine is called the Stack will store the program counter.

- b) A Subroutine is a set of instructions that can be called from within the main loop of a program.

The first reason why they are used is to make the program much shorter, simple and easier to read. This is due to not having the code from within the Subroutine in your main program. You can just have 'Call' and the name of the subroutine.

The second reason they are used is due to the fact you can call the same routine numerous times in a program. This means you only have to write the Subroutine once and call it whenever you need it instead of writing the code for the subroutine several times.

The third reason you would use a subroutine is due to the fact it is easier to rectify mistakes. If you have a subroutine you only have to correct this and the program will call the corrected version. Whereas if you have written it out you will need to correct

Connor Daniel Sheet 3

it every time you have used it in your program.

- c) Interrupts can occur at any time. Once an interrupt occurs the processor will store the program counter and set the flag. This will allow it to jump to the interrupt vector.

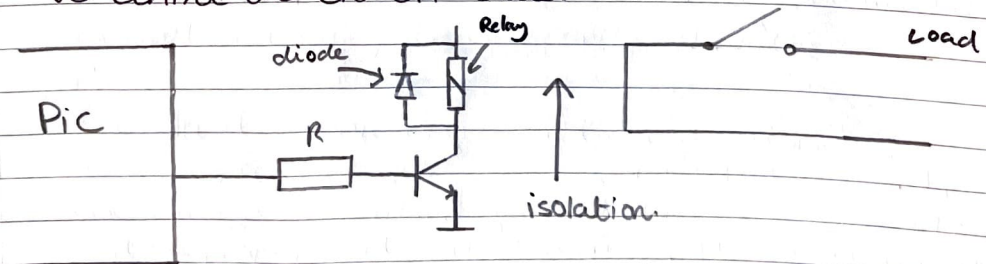
To ensure correct program operation following the interrupt it will save the W register and save the status. The ISR will then be completed.

Once the ISR has been completed, the W register and status will be restored to ensure the program returns to the correct state. The last action that will be necessary will be to clear the flag. If the flag is not clear you will be stuck with in the Interrupt routine and will not be able to return to the main program.

Question 3

- a) The PIC only produces D.C output signals. There is also limitations on the amps on the output pins of 25mA.

- b) I would use a relay and contactor for a PIC to control the ON OFF state.



If you have a big load for a PIC such as 230v you would use a transistor. Then you would have a relay, the relay contact would drive the load. This provides isolation between the load and the PIC. The transistor can switch the relay which can then switch a relatively large current. If you are switching a relay with a transistor it is important to use a diode. This is due to the fact the diode will remove any negative voltage from the coil inside the relay being an inductor.

c) The issue with a mechanical switch is that the contact can bounce. This means the switch will make, break and then make again. This means you will need a 'debounce routine' in order to correctly read the switch. Another method you could use would be an opto isolator. This provides electrical isolation between the PIC and the load by the use of a diode (LED) and an optic ~~isolator~~ transistor. The diode will turn on when supplied with power. This light from the diode will shine on to the base of the transistor. As it is an optic transistor it will turn on due to the photons. When the transistor is on the voltage of ~~the~~ above the resistor will go low to 0v, providing 5v to the PIC. When the transistor is off the voltage of the resistor goes back up to 5v.

Connor Daniel sheet 5

d) $12 \text{ bit} = 2^{12} = 4096$
meaning there are 4095 steps

$$\text{each step} = \frac{10}{4095} = 0.002442$$

$$\frac{6.5}{0.002442} = 2661.752 = 2662$$

Binary =

2048	1024	512	256	128	64	32	16	8	4	2	1
1	0	1	0	0	1	1	0	0	1	1	0

$2662 - 2048$
 $= 614$

$614 - 512$ $= 102$	Binary of 2662 (6.5v) $= 101001100110$
------------------------	---

$$102 - 64$$
$$= 38$$

$$38 - 32$$
$$= 6$$

$$6 - 4$$
$$= 2$$

$$2 - 2 =$$
$$0$$

Question 4

a) $2^{13} = 8192$

$8192 = 8$

1024

3 to 8 decoder truth table.

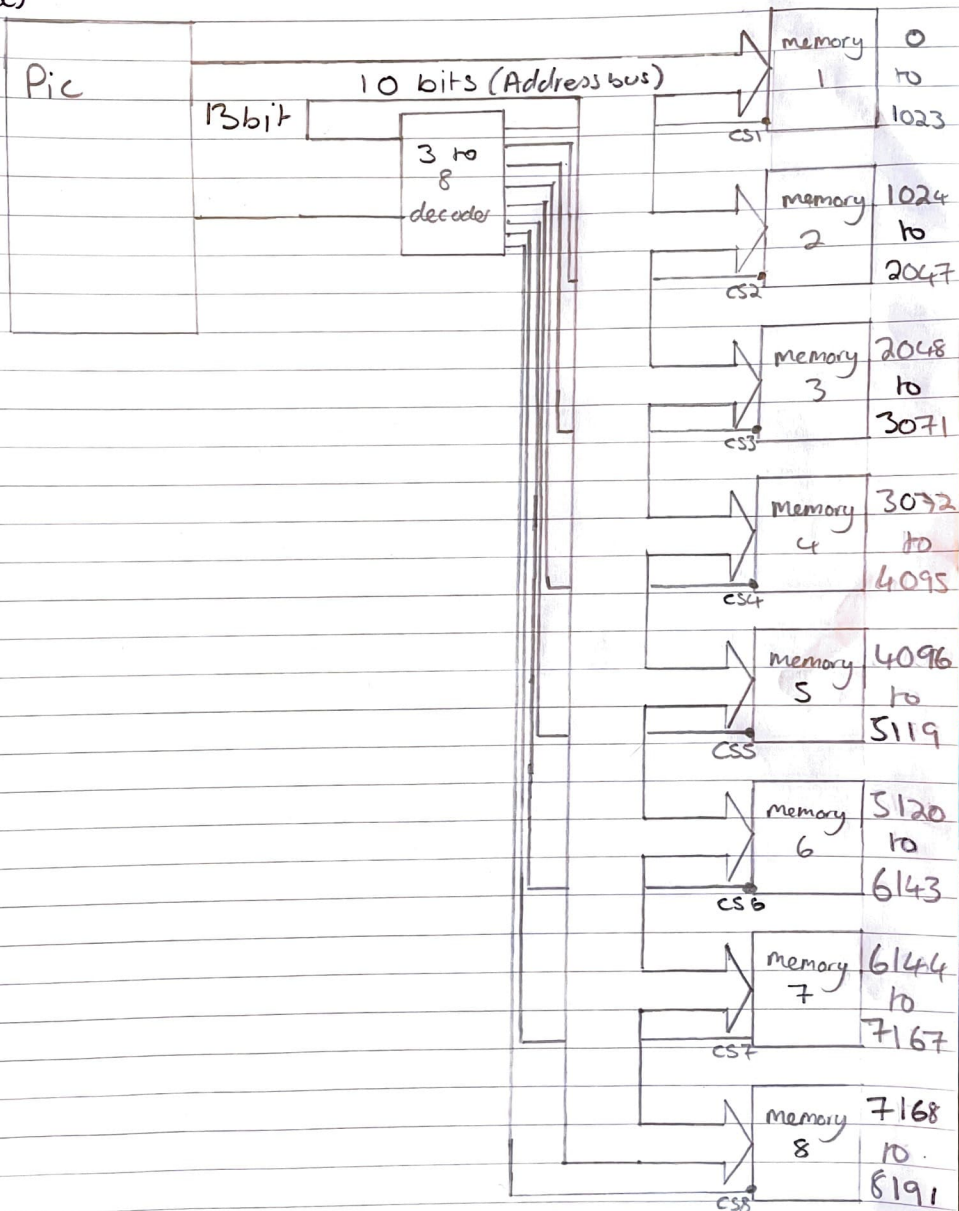
A10	A9	A8	CS8	CS7	CS6	CS5	CS4	CS3	CS2	CS1
0	0	0	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	1	1	0	1
0	1	0	1	1	1	1	1	0	1	1
0	1	1	1	1	1	1	0	1	1	1
1	0	0	1	1	1	0	1	1	1	1
1	0	1	1	1	0	1	1	1	1	1
1	1	0	1	0	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1	1

Connor Daniel

Sheet 67

Question 4

a)



Connor Daniel Sheet 8

b)

	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
~~4699~~	1	0	0	1	0	0	1	0	1	1	0	1	1

$4699 - 4096$
 $= 603$

$4699 =$

$603 - 512$

$= 91$

100		1001011011
This indicates		$512 + 64 + 16 + 2 + 1$
CSS		$= 603$

$91 - 64$

$= 27$

CSS base address = 4096

$27 - 16$

$= 11$

~~4096 +~~

$4096 + 603 = \underline{4699}$

$11 - 8$

$= 3$

~~3~~ - 2

$= 1$

$1 - 1 = 0$

- c) The difference ^{is} between Von neuman and Harvard architectures ~~are~~ that Von neuman will have the same physical ~~address~~ ^{memory} for the instructions and the data. However Harvard will use separate memory addresses for instructions and data. This means. For von neuman there is ~~separate~~ a common bus for data and instruction transfer. Where as for Harvard separate buses are used for transferring data and instruction.

Another difference would be that Von neuman takes two ~~clock~~ cycles to execute a single instruction. Harvard will execute an instruction in one cycle.

Question 5

- a) The three modes of transfer for DMA are:
- Burst mode - In this mode the DMA takes control until it has finished the data transfer. The CPU is inactive during this time.
 - Cycle stealing mode - In this mode the DMA controller requests control. The CPU grants control when it is not using the system buses. The DMA controller transfers 1 byte then releases the bus. It then requests access again for the next byte and so on. The CPU retains control but the transfer is slower.
 - Transparent mode - In this mode the DMA controller effectively monitors the buses to see when they are free and transfers data independantly. The CPU is unaware of this transfer.

- b) An asynchronous serial interface will transmit data over a physical medium such as a UART. An synchronous serial interface will use a clock to transmit data such as in I^2C .

- c) I^2C will use two lines and multiple slaves will connect to them whereas SPI will need to add a slave select line for every slave.

SPI is faster than I^2C

SPI is half duplex whereas I^2C is full duplex.

- d) To start a message the master will pull the SDA line low while the SCL line is still high. To end a message the master will pull the SDA line high when the SCL line is high.