UNIVERSITY OF GLASGOW
Glasgow College UESTC

Degree of BEng in Electronic and Electrical Engineering

Embedded Processors (UESTC2004)

Final Exam

Tuesday 26th June 2018
10.00 – 12.00

The numbers in square brackets in the right-hand margin indicate the marks allotted to the part of the question against which the mark is shown.

Attempt all questions

An electronic calculator may be used provided that it does not have a facility for either textual storage or display, or for graphical display.

Q1.

(a) Convert the value of 0011011011001101 from binary to hexadecimal. **[4]**

(b) Calculate the Boolean logical operations in Table Q1 below: **[10]**

Table Q1

A	B	A&B	~(A&B)	A|B	A^B
0	0				
0	1				
1	0				
1	1				

(c) What are the basic components of a computer? **[2]**

(d) Contrast the methods used by Harvard and von Neumann architecture microcontrollers to transfer information between CPU and memory, illustrating your discussion with a sketch of each? **[8]**

(e) Describe register pipeline, and the advantages and disadvantages of its use in digital systems? **[8]**

(f) State the pipeline stages used in the Cortex-M3 for instruction executions? **[6]**

Q2.

(a) Explain the purpose of a digital to analogue convertor (DAC). **[4]**
List three figures of merit important in a DAC, and explain how the value of each one is calculated? **[4]**

(b) The following code has been written to make use of an analogue input on an LPC1768 microcontroller but has not been commented. Add appropriate comments to the code of Figure Q2? **[4]**

```
#include “mbed.h”
PwmOut PWM1(p21);
AnalogIn Ain(p20);
int main(){
while(1){
PWM1.period(0.010);
PWM1 = Ain;
wait(0.1);
}
}
```

Figure Q2

Q3. Answer the following questions related to serial communications.

(a) Read the following C program, Figure Q3, which is executed on an mbed board, and answer the following questions.

```
#include "mbed.h"
Serial async_port(p9, p10);
char y;

int main()
{
        async_port.baud(19200);
        async_port.format(8, Serial::Even, 1);
        y = 0x43;
        while(1)
        {
                async_port.putc(y);
        }
        return 0;
}
```

Figure Q3

i. Sketch the waveform as observed at pin 9 of the mbed board. Show the transmission of one character. Describe the order in which the bits are transmitted. **[7]**

ii. Calculate the throughput of the UART. **[3]**

iii. If a framing error occurs due to incorrect capture of the stop bit, calculate the percent of the baud rate mismatch between the receiving UART and transmitting UART. **[5]**

(b) Answer the following questions related to SPI communications.

i. Briefly describe the main differences between UART and SPI. **[5]**

ii. The value 'y' (i.e., 0x71) is now transmitted from the mbed board through a SPI port. If the clock rate is 20MHz, what are the transmitted bits and their sequence at the data output pin of the board? How long will it take to transmit the character? **[5]**

Q4. Answer the following questions.

(a) The input of a 10-bit digital counter is connected to a clock signal. An interrupt occurs at the moment of overflow. The frequency of the clock signal is 512kHz, and the digital counter is initialized to zero.

i. Calculate the value of the interrupt frequency. **[6]**

ii. Explain how this digital counter can be used to generate two PWM outputs. **[4]**

(b) Which of the following statements is correct? Also explain why the other three answers are wrong. **[3]**

1) An interrupt service routine (ISR) is supposed to disable the non-maskable interrupt.
2) An interrupt service routine (ISR) is supposed to restore the context and return.
3) An interrupt service routine (ISR) is supposed to restore the lowest-priority interrupt.
4) An interrupt service routine (ISR) is supposed to increase the program counter.

(c) Read the following C program, Figure Q4, which is executed on an mbed board, and answer the following questions.

```
#include “mbed.h”

InterruptIn button(p18);
DigitalOut led1(p5);
void ISR1(void);

int main()
{
     button.rise(&ISR1);
     while(1)
     {
          //Execute other program statements
     }

     return 0;
}

VOID ISR1()
{
     led1 = !led1;
}
```

Figure Q4

i. Explain how the interrupt is triggered and what will happen after the interrupt. **[4]**

ii. Explain the benefits of using hardware interrupt. **[3]**

(d) Explain how the CPU interacts with its peripherals using control registers. **[5]**