HKU SPACE Community College
CCIT4064 Microcontrollers (MCU)
Assignment 2

Name: _____ SID: _____ CL _____

Total Mark: ____ / 40

Remark: Unless otherwise specified, all questions should assume the Intel MCS-51 instruction set for assembly language questions and the Arduino Sketch (C/C++-based language) for high-level language or Arduino-related questions.

Section A – Multiple Choice (10 Marks) (1 Mark Each)

1. What is the result in the accumulator after the following instructions are executed?

```
MOV   A, #55H
AND   A, #0FH
```

A) 05H
B) 55H
C) 50H
D) Invalid code

2. A classic 8051 has 4 KB internal program memory and is connected to an external 64 KB ROM. The firmware image is 6 KB. Assume the external ROM is correctly decoded for the full 64 KB program space. Which statement is correct?

A) With $\overline{EA}$ = HIGH, the CPU fetches all 6 KB from internal ROM; with EA = LOW, it uses internal up to FFF_H and external above that.
B) With $\overline{EA}$ = HIGH, the CPU fetches the first 4 KB from internal ROM and the remaining 2 KB from external; with EA = LOW, the CPU fetches all 6 KB from external.
C) With EA = HIGH, the CPU fetches everything from external; with EA = LOW, it fetches everything from internal.
D) The EA pin only affects external data memory; program fetch is unaffected.

3. Will this code segment be executed continuously?

```
LOOP:   MOV R0, #01H
        DJNZ R0, LOOP
```

A) Yes
B) No
C) It depends on the Carry bit.
D) Invalid code

4. Which of the following is directly interpreted and executed by the CPU hardware without translation?

A) C language
B) Assembly language
C) Machine code
D) Java

5. Which of the following instructions uses indirect addressing mode in MCS-51?

A) MOV A, 30H
B) MOV A, #30H
C) MOV A, @R1
D) MOV A, R1

6. Find the number of machine cycles the following loop will be executed:

```
BACK:   MOV   R6, #200
        MOV   R5, #200
HERE:   DJNZ  R5, HERE
        DJNZ  R6, BACK
        END
```

A) ~20000
B) ~40000
C) ~80000
D) Infinite

---

7. Based on the principles of flame detection discussed in the article written by Winsen Electronics (https://www.winsen-sensor.com/knowledge/principle-of-flame-detection.html), which of the following BEST explains why a simple broadband IR phototransistor such as YG1006 (used in Lab03) is rarely used alone in professional industrial flame detection systems?

I Industrial detectors often rely on narrow-band spectral discrimination to target specific combustion emission signatures.
II Reliable flame detection frequently requires temporal pattern analysis to distinguish flames from steady IR sources.
III Multi-sensor fusion improves immunity to environmental interference.
IV The primary limitation of YG1006 is insufficient infrared sensitivity rather than lack of spectral discrimination.

A) I and II only
B) I, II and III only
C) I, II and IV only
D) I, II, III and IV

---

8. According to the Arduino UNO Q (2GB) reference design, which LPDDR4 memory device is used on the board?
   Hint: Students should use the Arduino UNO Q hardware documentation page as a starting point: https://docs.arduino.cc/hardware/uno-q/

A) Model: IME8G32L4HA; Package: 200-ball FBGA
B) Model: MT53E512M32D1; Package: 200-ball TFBGA
C) Model: MT53E512M64D2; Package: 376-ball WFBGA
D) Model: H9HCNNN8KUM; Package: 200-ball FBGA

---

9. Given firmware written in MCS-51 Assembly Language, which of the following microcontroller setups will execute the code the fastest?

A) Intel 8051 MCU running at 12.0000 MHz (12 clocks per machine cycle)
B) Company B's 8051-variant MCU running at 6.0000 MHz (6 clock per machine cycle)
C) Company C's 8051-variant MCU running at 8.0000 MHz (4 clock per machine cycle)
D) Company D's 8051-variant MCU running at 24.0000 MHz (1 clock per machine cycle)

---

10. Which ports on the Intel 8051 act as the 16-bit address lines for transferring data through it?

A) P0 and P1
B) P1 and P2
C) P2 and P3
D) None of the above

A1	A2	A3	A4	A5
A6	**A7**	**A8**	**A9**	**A10**

Section B (Short Question) (Total 30 Marks) (*Answer the questions in the space provided below*)

1) The original Intel MCS-51 microcontroller architecture was designed in the 1980s. In past decades, many modern derivatives of the 8051 architectures were built; this question investigates part of the many improvements and benefits of the modern technologies integrated into them.

(a) Identify the instruction(s) that take the longest execution time on a traditional MCS-51 (8051). How many clock cycles (oscillator periods) do they take?

(1 Mark)

(b) If a traditional MCS-51 variant runs at 30 MHz, how long does it take to execute that instruction?

(2 Marks)

(c) With reference to the Silicon Laboratories modern MCS-51 variant, the EFM8UB1 (see the EFM8UB1 Reference Manual: https://www.silabs.com/documents/public/reference-manuals/efm8ub1-rm.pdf), which is based on the CIP-51 core, a fully software-compatible enhanced implementation of the standard MCS-51 instruction set, determine the clock cycles required to execute the same instruction(s) identified in Part (a), assuming the Prefetch ON configuration.

(1 Mark)

(d) If the EFM8UB1 runs at 30 MHz, how long does it take to execute that instruction?

(2 Marks)

(e) At the same frequency (30 MHz), what is the **minimum** performance improvement factor (how many times faster) for EFM8UB1 vs traditional MCS-51?

(1 Mark)

2) Write an 8051 assembly language program to perform the following operations. (The **ORG** and **END** directives are optional.)
   - **(a)** Load the accumulator (**A**) with the value 80_H.
   - **(b)** Load register **R0** with the value 64_H. (Label this instruction as "**Q1Lable**".)
   - **(c)** Multiply the accumulator by 128. (Hint: The multiplication result is stored as: the high byte in register B and the low byte in accumulator A.)
   - **(d)** Add the contents of **R0** to the accumulator.
   - **(e)** Move the content of the accumulator (**A**) to the low-byte of **DPTR**.
   - **(f)** Move the content of register **B** to the high-byte of **DPTR**.

   Note: The complete program (including starting and ending directives) should be within 10 lines. Remark: The program must use only native Intel MCS-51 instructions. Do not use macros, pseudo-instructions, or IDE/assembler shortcuts that expand into multiple instructions. To avoid assembler symbol assumptions, write all SFR accesses (except A/ACC) using direct addresses.

(4 Marks)

```
                ;
_____________   _____________________________
                ;
_____________   _____________________________
                ;
_____________   _____________________________
                ;
_____________   _____________________________
                ;
_____________   _____________________________
                ;
_____________   _____________________________
                ;
_____________   _____________________________
                ;
_____________   _____________________________
                ;
_____________   _____________________________
                ;
_____________   _____________________________
                ;
_____________   _____________________________
```

3) Following Question B2, what is the address indicated by DPTR?

(2 Marks)

4) By default, the Intel MCS-51 instruction set does not provide an **ADD B, Rn** instruction. You are required to perform the equivalent operation in 8051 assembly. Write a subroutine named ADDB that performs the following:
   - Adds the value in register R0 to register B, and stores the result back in B (i.e., B ← B + R0).
   - Preserves the original contents of A (ACC) and the PSW register.
   - The subroutine must start at address 0100H.
   - Write brief comments for each instruction to explain your approach.

   Assume that the first two bytes of general-purpose internal RAM, 30H and 31H, are available for temporary storage.
   Note: The complete subroutine (including starting/ending directives) must be within 10 lines.
   Remark: Assembler symbols (e.g., A, B, PSW) may be used.

(5 Marks)

```
                ;
                ;
                ;
                ;
                ;
                ;
                ;
                ;
                ;
                ;
                ;
```

5) Calculate the results (both binary and hexadecimal values) with the calculation step and show the status of CY, AC, and P flags after the addition of $(AA)_{16}$ and $(88)_{16}$ of the following instructions.

MOV A, #AAH
ADD A, #88H

(4 Marks)

6) You are required to use an Arduino UNO R3 to control the brightness of an LED using a potentiometer as the analog input.

(a) Draw a simplified schematic diagram and clearly indicate the wiring connections among the following components:

- Arduino UNO R3
- 1 × LED
- 1 × potentiometer
- 1 × current-limiting resistor

Your circuit must be able to:

- Read an analog voltage from the potentiometer using an Arduino analog input pin.
- Convert the measured analog input level into an appropriate output control signal for adjusting LED brightness.

Remark: For simplicity, you may ignore pins that are not used in your design (e.g., AREF).

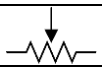
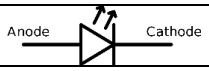
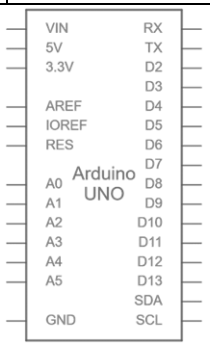
		
1 x Resistor	1 x Potentiometer	1 x LED
 1 x Arduino UNO R3		

Table 1: Component List for Question 6

(3 Marks)

(b) Write a **complete Arduino sketch** that implements the circuit in (a). Your program must achieve all of the following:

- Analog Input Reading and Level Conversion
  - Read the potentiometer value using analogRead() (range: 0 to 1023).
  - Convert the analog input into **five discrete levels**: **0, 1, 2, 3, 4**, according to the following ranges:

Analog Input Range	Level
0 – 200	0
201 – 400	1
401 – 600	2
601 – 800	3
801 – 1023	4

- Serial Monitor Output
  - Print the analog input value and its level to the Serial Monitor in the following format:

Analog Input: <value>; Level: <level>
(Example: Analog Input: 356; Level: 1)

- LED Brightness Control
  - Control the LED brightness based on the converted level.
  - Convert the analog input into **five discrete levels**: **0, 1, 2, 3, 4**, according to the following ranges:
    - **0** → LED **OFF**; **1** → LED at **1/4 intensity**; **2** → LED at **half intensity**; **3** → LED at **3/4 intensity**; **4** → LED at **full intensity**

(5 Marks)

Important remarks:

- Answer all questions in the space provided.
- Submit your assessment via SOUL. **Do NOT compress your files.**
- Submission should include the following file:
  - CCIT4064_A2_**YourName**_**SID**.pdf (convert your completed answer book to PDF format)
- Penalties will apply in (but are not limited to) the following cases:
  - Assignment submitted after the grace period.
  - Assignment submitted with an incorrect file format or file name.
  - Missing or incorrect student information in the assignment.
- This is an INDIVIDUAL assessment. Plagiarism or collusion, once verified, will result in mark penalties and an unofficial record.

- End of Assignment 2 -