KEY # CSC 230 AI made Practice Exam 2 Fall 2024
Practice Exam #2

Students must check the number of pages in this examination paper before beginning to write, and report any discrepancy immediately.

Total marks: 75
Time allowed: 2 hours

Section A (30 marks): For each question in this section, place an X beside all answers that apply. Each question is worth 6 marks. Partial marks are not given for incomplete answers.

Question 1: For a 7-bit two's complement number in AVR:
_X_ The maximum positive value is 0b0111111
_X_ The most negative value is 0b1000000
___ Zero can be represented as 0b0000000 or 0b1000000
_X__ The range of values is -64 to +63
___ None of the above

Question 2: Regarding AVR memory addressing:
_X__ The Z register can be used for indirect jumps
_X_ The Y register consists of r29:r28
_X_ Post-increment addressing updates the pointer after accessing memory
_X__ ld and st instructions can only use X, Y, or Z registers
___ None of the above

Question 3: In the context of AVR I/O ports:
___ Writing 1 to a DDRD bit makes that pin an input
___ Pull-up resistors are enabled by default
_X__ The PORTD register can be read and written
_X_ Pin states can be read regardless of direction setting
___ None of the above

Question 4: The AVR status register:
_X__ Is automatically saved during interrupt handling
_X__ Contains the global interrupt enable bit
_X__ Is modified by arithmetic and logic instructions
_X_ Can be directly copied to a general-purpose register
___ None of the above

Question 5: For AVR branch instructions:
_X__ Branch targets must be within -64 to +63 words of the branch
___ Multiple conditions can be tested in a single branch
_X__ rjmp has a greater range than breq
_X__ Branch instructions are two bytes long
___ None of the above

Section B (25 marks): Short Answer Questions

Question 6 (5 marks):
Explain how the AVR hardware stack operates during an interrupt service routine, including register preservation and return address handling.

When an interrupt occurs:
1. Program counter pushed onto stack
2. Global interrupt flag cleared
3. Status register automatically pushed
4. Jump to interrupt vector
5. ISR must save/restore used registers

Question 7 (5 marks):
Describe the purpose of the AVR status register's H (half-carry) flag and when it becomes relevant in arithmetic operations.

H flag indicates carry from bit 3 to bit 4. Used in BCD arithmetic operations. Set when carry occurs from lower nibble to upper nibble during addition, or when borrow occurs during subtraction. Essential for decimal arithmetic adjustments.

Question 8 (5 marks):
Compare direct and indirect addressing modes in AVR assembly, explaining their advantages and typical use cases.

Direct addressing: Immediate operand access, fixed memory locations, simple and fast.
Indirect addressing: Flexible pointer-based access, arrays/data structures, dynamic memory access.

X/Y/Z registers enable pre/post increment/decrement modes.
Direct for static data, indirect for arrays and dynamic structures.

Question 9 (5 marks):
Explain how the AVR processor handles the overflow flag (V) in signed arithmetic operations. Provide specific examples.

V flag set when signed overflow occurs:
- *Addition: operands same sign, result opposite sign*
- *Subtraction: operand signs different, result same sign as subtrahend*

Used with signed branch instructions (BRLT, BRGE)
Critical for signed arithmetic validity checking

Question 10 (5 marks):
Describe the function and importance of the linker command file in AVR program development.

Linker command file
- *Defines memory layout and sections*
- *Maps program sections to memory regions*
- *Specifies start address and stack location*
- *Controls placement of code and data segments*
- *Essential for proper memory allocation*

Section C (20 marks): Assembly Programming Problems

Question 11 (4 marks):
Write AVR assembly code to swap the contents of two memory locations without using any general-purpose registers.

```
lds r16, ADDR1      ; Load first value
lds r17, ADDR2      ; Load second value
sts ADDR1, r17      ; Store second value to first address
sts ADDR2, r16      ; Store first value to second address
```

Question 12 (4 marks):
Write code to configure Timer1 for PWM operation on OC1A (Port B, pin 1) with a frequency of 50Hz.

```
ldi r16, (1<<WGM11)|(1<<COM1A1)
sts TCCR1A, r16
ldi r16, (1<<WGM12)|(1<<CS11)|(1<<CS10)
sts TCCR1B, r16
ldi r16, HIGH(ICR1_VALUE)
sts ICR1H, r16
ldi r16, LOW(ICR1_VALUE)
sts ICR1L, r16
```

Question 13 (4 marks):
Write an interrupt service routine for External Interrupt 0 that toggles all bits in PORTB while preserving all register values.

```
ISR_INT0:
   push r16
   in r16, SREG
   push r16
   in r16, PORTB
   com r16
   out PORTB, r16
   pop r16
   out SREG, r16
   pop r16
   reti
```

Question 14 (4 marks):
Write assembly code to perform a 16-bit addition of values stored in SRAM at addresses DATA1 and DATA2, storing the result at address RESULT.

```
lds r16, DATA1
   lds r17, DATA1+1
   lds r18, DATA2
   lds r19, DATA2+1
   add r16, r18
   adc r17, r19
   sts RESULT, r16
   sts RESULT+1, r17
```

Question 15 (4 marks):
Write code to rotate an 8-bit value in r16 right by the number of positions specified in r17, preserving the original value in r16.

```
mov r18, r16          ; Save original value
loop:
   tst r17            ; Check count
   breq done          ; Exit if zero
   lsr r16            ; Rotate right
   dec r17            ; Decrement count
   rjmp loop          ; Repeat
done:
   nop
```