#HW, cec32x_HW_34–HW8_conditional_execution, 140 points total.

Soln to HW 8. Conditional Execution

Prob 1. (40 points total) Consider a 6-bit system with the following 4 cases: 1) r0 = 22, r1 = 20, 2) r0 = 20, r1 = 22, 3) r0 = -20, r1 = -22, and 4) r0 = 20, r1 = -22. Determine the condition flags Z, N, C, V after executing `CMP r0, r1`.

Soln:

```
1) (a) Carry flag for unsigned: 22 - 20 = 2 \in [0, 65],
       ==> B = 0 ==> C = 1
   (b) Overflow flag For signed: 22 - 20 = 2 \in [-32, 31],
       ==> V = 0
   (c) Zero flag: result is not 0 ==> Z = 0
   (d) Negative flag: result is not negative ==> N = 0

2) (a) Carry flag for unsigned: 20 - 22 = -2 \notin [0, 65],
       ==> B = 1 ==> C = 0
   (b) Overflow flag For signed: 20 - 22 = -2 \in [-32, 31],
       ==> V = 0
   (c) Zero flag: result is not 0 ==> Z = 0
   (d) Negative flag: result is negative ==> N = 1

3) (a) Carry flag for unsigned: (-20) - (-22)
       ==> (64-20) - (64-22) = 2 \in [0, 65], ==> B = 0 ==> C = 1
   (b) Overflow flag For signed: (-20) - (-22) = 2 \in [-32, 31],
       ==> V = 0
   (c) Zero flag: result is not 0 ==> Z = 0
   (d) Negative flag: result is not negative ==> N = 0

4) (a) Carry flag for unsigned: 20 - (-22)
       ==> (20 - (64-22) = -22 \notin [0, 65], ==> B = 1 ==> C = 0
   (b) Overflow flag For signed: 20 - (-22) = 42 \notin [-32, 31],
       ==> V = 1
   (c) Zero flag: result is not 0 ==> Z = 0
   (d) Negative flag: MSB of result is 1 ==> N = 1
       Or 42 ==> 42-64 = -22 which is negative ==> N = 1
```

Prob 2. (20 points total) Use the condition flag results of the example in Section 2.2 of class notes for Module 8 for the following assignments.

- For signed numbers r0 = 12, r1 = 10, both GE and GT (`CMP r0, r1`) should be true, and both LE and LT should be false. Determine the results of the following (AND and

OR logic operations) to verify the above claim of the condition code (GE and GT are true, and LE and LT are false):

  - LT: $N\bar{V} + \bar{N}V$
  - LE: $Z + N\bar{V} + \bar{N}V$
  - GE: $NV + \bar{N}\bar{V}$
  - GT: $\bar{Z}(NV + \bar{N}\bar{V})$

- For unsigned numbers r0 = 12, r1 = 10, both HS and HI (`CMP r0, r1`) should be true, and both LS and LO should be false. Determine the results of the following to verify the above claim of the condition code (HS and HI are true, and LS and LO are false):

  - LO: $\bar{C}$
  - LS: $\bar{C} + Z$
  - HS: C
  - HI: $C\bar{Z}$

Soln:

For r0 = 12, r1 = 10, after `CMP r0, r1`, we have C = 1, V = 0, Z = 0, and N = 0, leading to

- For signed numbers:
  - LT: $N\bar{V} + \bar{N}V = 0 \cdot 1 + 1 \cdot 0 = 0$ => LT is false
  - LE: $Z + N\bar{V} + \bar{N}V = 0 + 0 = 0$ => LE is false
  - GE: $NV + \bar{N}\bar{V} = 0 \cdot 0 + 1 \cdot 1 = 1$ => GE is true
  - GT: $\bar{Z}(NV + \bar{N}\bar{V}) = 1 \cdot 1 = 1$ => GT is true
- For unsigned numbers:
  - LO: $\bar{C} = 0$ => LO is false
  - LS: $\bar{C} + Z = 0 + 0 = 0$ => LS is false
  - HS: $C = 1$ => HS is true
  - HI: $C\bar{Z} = 1 \cdot 1 = 1$ => HI is true

Prob 3. (20 points total) Use the condition flag results of the example in Section 2.2 of class notes for Module 8 for the following assignments.

- For signed numbers r0 = 10, r1 = -12, both GE and GT (`CMP r0, r1`) should be true, and both LE and LT should be false. Determine the results of the following (AND and OR logic operations) to verify the above claim of the condition code:

  - LT: $N\bar{V} + \bar{N}V$
  - LE: $Z + N\bar{V} + \bar{N}V$
  - GE: $NV + \bar{N}\bar{V}$
  - GT: $\bar{Z}(NV + \bar{N}\bar{V})$

- For unsigned numbers r0 = 10, r1 = -12 (-12 + 32 = 20), both HS and HI (`CMP r0, r1`) should be false, and both LS and LO should be true. Determine the results of the following to verify the above claim of the condition code:
    - LO: $\bar{C}$
    - LS: $\bar{C} + Z$
    - HS: C
    - HI: $C\bar{Z}$

Soln:

For r0 = 10, r1 = -12, after `CMP r0, r1`, we have C = 0, V = 1, Z = 0, and N = 1, leading to

- For signed numbers:
    - LT: $N\bar{V} + \bar{N}V = 1 \cdot 0 + 0 \cdot 1 = 0$ => LT is false
    - LE: $Z + N\bar{V} + \bar{N}V = 0 + 0 = 0$ => LE is false
    - GE: $NV + \bar{N}\bar{V} = 1 \cdot 1 + 0 \cdot 0 = 1$ => GE is true
    - GT: $\bar{Z}(NV + \bar{N}\bar{V}) = 1 \cdot 1 = 1$ => GT is true
- For unsigned numbers:
    - LO: $\bar{C} = 1$ => LO is true
    - LS: $\bar{C} + Z = 1 + 0 = 1$ => LS is true
    - HS: $C = 0$ => HS is false
    - HI: $C\bar{Z} = 0 \cdot 1 = 0$ => HI is false

Prob 4. (30 points total) Given the following C code for returning a 16-bit or 32-bit value using a pointer.

```
int32_t x_cmp_y_load_c(int32_t x, int32_t y, int16_t *ptrA,
        int32_t *ptrB) {
    if (x > y) {
        return *(ptrA+1);
    } else {
        return *(ptrB+2);
    }
}
```

Implement the above C code in assembly with the following start code.

```
x_cmp_y_load_s  PROC
                CMP     r0, r1
                Bxx     cmp_then
cmp_else        ....
                ....
cmp_then        ....
                ....
```

```
cmp_end         BX      lr
                ENDP
```

where `Bxx` should be replaced by the appropriate instruction and `....` can be 0 to multiple lines of assembly instructions.

Soln:

```
x_cmp_y_load_s  PROC
                CMP     r0, r1
                BGT     cmp_then
cmp_else        LDR     r0, [r3, #8]
                B       cmp_end
cmp_then        LDRSH   r0, [r2, #2]
cmp_end         BX      lr
                ENDP
```

Prob 5. (30 points total) Repeat Prob 4 assuming all the numbers, including the arguments and return, are unsigned instead of signed.

Soln:

```
x_cmp_y_load_s  PROC
                CMP     r0, r1
                BHI     cmp_then
cmp_else        LDR     r0, [r3, #8]
                B       cmp_end
cmp_then        LDRH    r0, [r2, #2]
cmp_end         BX      lr
                ENDP
```