Practical 3

Total marks: 10

Date: February 15, 2024

Assessment date: February 29, 2024

Instructions:

1. **This is a proctored practical.**
2. **Use Lab computer for writing, editing, debugging and executing your code.**
3. **You are allowed to use internet and class notes / slide decks.**
4. **You should demonstrate completed assembly code during the Lab session. Your work will be assessed by the lab tutors. If you fail to demonstrate, you will get zero mark for the assignment.**
5. **Upload completed Practical files (\*.S68) (create a ZIP file) to Turnitin. The Turnitin link has been provided at Blackboard.**

Edit, compile and execute the code across and observe contents of memory and registers.

|  |  |
| --- | --- |
|  |  |
| 1 | move.l #$00000492, d0  cmpi.w #$0020,d0  bge GE  move #$0000,d0  GE:  add d0,d3 |
| 2 | move.l #$00000002, d0 ; dec 2  cmpi.w #$8001,d0 ; dec -32767  bge GE  move #$0000,d0  GE:  add d0,d3 |
| 3 | move.l #$0001AABB20, d0  cmp.b #$00, d0  beq.s ValueIs20  move.b d0, d1    ValueIs20:  move.b d0, d2 |
| 4 | move.l #$FF940F21, d0  tst.w d0  beq LabelEq  move d0, d1  LabelEq:  move d0,d3 |
| 5 | MOVE.L #$9, D0  DIVU.W #$0002, D0    MOVE.L #$FFFFFFF5, D1  DIVS.W #$0002, D1 # check remainder and quotient. |
| 6 | # EXT performs a signed extension, as the top bit of a value is essentially an indicator of whether it is positive or negative  MOVE.L #$FEDCBA14, D0    EXT.W D0 ; 14 -> 0001 0100 ; msb positive (0), this is extended to 0000 0000 0001 0100  ; FEDCBA14 -> FEDC0014    MOVE.L #$C1208489, D1    EXT.L D1 ; 8489-> 1000 0100 1000 1001 ; |
| 7 | MOVE.B #100, $3000    MOVE.B #12, D2  MOVE.B #22, D3    AND.B D2, D3    LEA $3000, A1    SUB.B #20, (A1)    ADD.B #10, (A1) |
| 8 | move #3, d0 \*task #3 in D0  move.w #100, d1  trap #15 |
| 9 | move.l #$FF940000, d0  tst.w d0  beq LabelEq  move d0, d1  LabelEq:  move d0,d3 |
| 10 | move.l #$7FF290F5, d0  btst.l #$11, d0  beq LabelEq  move d0, d1  LabelEq:  move d0,d3 |
| 11 | START:  X EQU 3  Y EQU 7    MOVE #X, D0  MOVE #Y, D1    LOOP:  ADD #$1, D0  SUB #$1,D1  CMP D0, D1  BLE EXIT  BRA LOOP    EXIT:  MOVE #03, D0  TRAP #15  END START ; last line of source |
| 12 | ; nested loops; outer loop - infinte  INFLOOP:  move.b #$0F, $000400    DELAY:  move.l #$3, d6    DEL1:  sub.l #$1, d6    bne DEL1    move.l #$4, d0    jmp INFLOOP |
| 13 | INITIALISE:  MOVE.B #3, D0  MOVE.B #0, D1    PLAY\_LOOP:  CMP.B #1, D0  BNE GAME\_POINTS  BEQ GAME\_OVER    GAME\_POINTS:  ADD.B #10, D1  SUB.B #1, D0  BRA PLAY\_LOOP    GAME\_OVER: |