Quiz based on Lecture 4 and Lecture 5

Total points 9/12

The respondent's email address (**renny171647.cse@chitkara.edu.in**) was recorded on submission of this form.

✓ Which interrupt is never be disabled? 1/1

- ○ RST 2
- ○ RST 7.5
- ◉ TRAP ✓
- ○ RST 3

✓ Which of the following are comes under Software interrupts? 1/1

- ☑ RST 2 ✓
- ☐ RST 7.5
- ☐ TRAP
- ☑ RST 3 ✓

✓ Subroutine call comes under 1/1

- (●) Software interrupts ✓
- ( ) Internal interrupts
- ( ) Hardware interrupts
- ( ) External interrupts

✓ Which of the following are conditional branch instruction? 1/1

- ( ) BNZ
- ( ) BE
- ( ) BNC
- (●) All of the above ✓

✓ Which mnemonic is used to Add two Floating Point numbers? 1/1

- ( ) ADDI
- ( ) ADD
- (●) ADDF ✓
- ( ) ADDD

✓ Which of the following are bit manipulation and logical instruction? 1/1

- [x] Set carry (SETC) ✓
- [x] Enable interrupt (EI) ✓
- [x] Disable interrupt (DI) ✓
- [ ] None of the above

✓ Full form of PSW 1/1

- ○ Program System word
- ◉ Program Status Word ✓
- ○ Program Set Word
- ○ None of the above

✓ Types of Program Control instruction. 1/1

- ◉ Conditional & Unconditional ✓
- ○ Conditional
- ○ Unconditional
- ○ None of the above

✓ How many types of Data Manipulation and transfer instruction used in executing an instruction? 1/1

- ○ Four
- ◉ Three ✓
- ○ Two
- ○ One

✗ Explian Subroutine Call and Return. ···/3

Procedure is a self contained sequence of instructions that perform a given computational task branch at beginning of procedure and branch that to main programme and at end of procedure also called as subroutine. Store the value of the PC before proceeding with the procedure normally usess the memory stack to keep track of return address.

Feedback

Procedure Call and Return Instructions:
procedure is a self-contained sequence of instructions that performs a given computational task branch at beginning of procedure and branch back to main program at end of procedure also called a subroutine
stores the value of the PC (called the return address) before proceeding with the procedure normally uses a memory stack to keep track of the return address.
A procedure call would be implemented with the following microinstructions:
SP □ SP – 1 Decrement the stack pointer
M[SP] □ PC Store the return address on the stack
PC □ Effective Addr Transfer control to the procedure
Procedure return microinstructions would become (in correct order):
PC □ M[SP] Transfer return address to the PC
SP □SP + 1 Pop the Stack by incrementing the stack pointer