**Please explain how the stack is used for subroutine call and return (Chapter 7 “Stack Instructions” (Maximum 1 point)**

In the called subroutine, the first code executed is usually termed the subroutine prologue, since it does the necessary housekeeping before the code for the statements of the routine is begun.

For instruction set architectures in which the instruction used to call a subroutine puts the return address into a register, rather than pushing it onto the stack, the prologue will commonly save the return address by pushing the value onto the call stack, although if the called subroutine does not call any other routines it may leave the value in the register. Similarly, the current stack pointer and/or frame pointer values may be pushed.

If frame pointers are being used, the prologue will typically set the new value of the frame pointer register from the stack pointer. Space on the stack for local variables can then be allocated by incrementally changing the stack pointer.

References:

Call stack - Wikipedia. (n.d.). Retrieved October 30, 2021, from <https://en.wikipedia.org/wiki/Call_stack>.

**Explain a computer's register-level architecture, including: (Maximum 1 point)**

**a) CPU-memory interface (Your response to this question should be substantial (minimum 100 words). The explanations should be written (without copy and paste any graphs or diagrams)**

**b) special-use registers (minimum 100 words). The explanations should be written (without copy and paste any graphs or diagrams).**

**c) addressing modes (minimum 100 words). The explanations should be written (without copy and paste any graphs or diagrams).**

A.

CPU-Memory interface includes the data bus, address bus and some control signals including Read, Write, and Memory-Function-Complete (MFC). The CPU is interfaced to the data bus and address bus through the MDR and MAR registers, respectively.

In addition to this interface, there is a need for a CPU-Memory interface circuitry to manage their interaction. When the CPU wants to perform a read or write operation, it asserts either the Read or Write signal and puts the address to be read from or written to in the MAR register. Then, the CPU waits for the memory to finish the requested transfer operation. It is required that the CPU keeps the Read or Write signal set until the memory finishes the requested operation. The memory activates the MFC signal when the requested operation is completed. One the MFC is set to 1, and then the Read or Write signal can be set to 0. This interaction process between the CPU and memory is called handshaking.

References:

CPU-Memory Interface Circuit. (n.d.). Retrieved October 30, 2021, from <https://faculty.kfupm.edu.sa/COE/aimane/assembly/pagegen-172.aspx.htm>.

B.

A Special-Use Register is a register within a microprocessor, which controls or monitors various aspects of the microprocessor's function. Depending on the processor architecture

Because special registers are closely tied to some special function or status of the processor, they might not be directly writeable by normal instructions (such as adds, moves, etc.). Instead, some special registers in some processor architectures require special instructions to modify them. For example, the program counter is not directly writeable in many processor architectures. Instead, the programmer uses instructions such as return from subroutine, jump, or branch to modify the program counter. For another example, the condition code register might not directly writable, instead being updated only by compare instructions.

References:

Special function register - Wikipedia. (n.d.). Retrieved October 30, 2021, from <https://en.wikipedia.org/wiki/Special_function_register>.

C.

Addressing modes are an aspect of the instruction set architecture in most central processing unit (CPU) designs. The various addressing modes that are defined in a given instruction set architecture define how machine language instructions in that architecture identify the operand(s) of each instruction. An addressing mode specifies how to calculate the effective memory address of an operand by using information held in registers and/or constants contained within a machine instruction or elsewhere.

In computer programming, addressing modes are primarily of interest to those who write in assembly languages and to compiler writers.

In particular, different authors and computer manufacturers may give different names to the same addressing mode, or the same names to different addressing modes. Furthermore, an addressing mode which, in one given architecture, is treated as a single addressing mode may represent functionality that, in another architecture, is covered by two or more addressing modes.

References:

Addressing mode - Wikipedia. (n.d.). Retrieved October 30, 2021, from <https://en.wikipedia.org/wiki/Addressing_mode>.