|  |  |
| --- | --- |
| **Computer Architecture** | **Computer Organization** |
| deals with those attributes of a computer system that are visible to the programmer | deals with how different components of a computer are linked together to meet the requirements specified by the computer architecture. |
| abstract design of a computer system from programmer’s point of view | concrete implementation of the computer architecture from hardware designer’s point of view |
| deals generally with higher level design issues | deals generally with low level design issues |
| CA best describes what a computer system does and what components does the system consist of. | *CO best describes how are the components of computer system connected or wired and how will they communicate with each other.* |
| As an example, computer architecture concerns with how the operands are addressed in a multiplication instruction. | As an example, the computer organization concerns with whether the multiplication is done using a single multiplication unit or by repeated additions. |
| Architectural attributes of the computer system are the Instruction Set, addressing modes, physical memory, data formats, I/O mechanism etc. | Organizational attributes of the computer system are hardware details, peripheral devices, control signals, etc. |
|  |  |

|  |  |
| --- | --- |
| **Computer Architecture** | **Computer Organization** |
| Computer architecture refers to those attributes of a system visible to a programmer or, put another way, those attributes that have a direct impact on the logical execution of a program. | Computer organization refers to the operational units and their interconnections that realize the architectural specifications. |
| Examples of architectural attributes include the instruction set, the number of bits  used to represent various data types (e.g., numbers, characters), I/O mechanisms,  and techniques for addressing memory. | Organizational attributes include those  hardware details transparent to the programmer, such as control signals; interfaces  between the computer and peripherals; and the memory technology used. |
| For example, it is an architectural design issue whether a computer will have  a multiply instruction. | It is an organizational issue whether that instruction will  be implemented by a special multiply unit or by a mechanism that makes repeated  use of the add unit of the system. |

Solutions for barsha

2016 Spring (Computer Architecture)

Q. 1 a)

Y = (A+B\*C) / (E-F)

0 Address Format

PUSH-B

PUSH-C

MUL

PUSH-A

ADD

PUSH-E

PUSH-F

SUB

DIV

POP-Y

3 Address Format

MUL P, B, C ; P 🡨 [B] \* [C]

ADD Q, A, P ; Q 🡨 [A] + [P]

SUB R, E, F ; R 🡨 [E] – [F]

DIV Y, Q, R ;Y 🡨 [Q] / [R]

1 Address Format

LOAD E ; Acc 🡨 [E]

SUB F ; Acc 🡨 [Acc] - [F]

STORE P ;P 🡨 [Acc]

LOAD B ; Acc 🡨 [B]

MUL C ;Acc 🡨 [Acc] \* [C]

ADD A ; Acc 🡨 [Acc] + [A]

DIV P ; Acc 🡨 [Acc] / [P]

STORE Y ;Y 🡨 [Acc]

2 Address Format

MOV P, B ; P 🡨 [B]

MUL P, C ; P 🡨 [P] \*[C]

ADD P, A ; P 🡨 [P] +[A]

MOV Q, E ; Q 🡨 [E]

SUB Q, F ; Q 🡨 [Q] -[F]

DIV P, Q ; P 🡨 [P] / [Q]

MOV Y, P ; Y 🡨 [P]

The types of Registers in computer system are:

1. Memory Address Register
2. Memory Buffer Register
3. Instruction Buffer Register
4. Instruction Pointer
5. Instruction Register
6. Accumulator Register
7. Data Register
8. General Purpose Registers
9. Multplier Quotient Register