Questions :

3.1 What general categories of functions are specified by computer instructions?

- **Processor-memory**: Data may be transferred from processor to memory or from memory to processor.
- **Processor-I/O**: Data may be transferred to or from a peripheral device by transferring between the processor and an I/O module.
- **Data processing**: The processor may perform some arithmetic or logic operation on data.
- **Control**: An instruction may specify that the sequence of execution be altered.

3.2 List and briefly define the possible states that define an instruction execution.

- **Instruction address calculation (iac)**: Determine the address of the next instruction to be executed.
- **Instruction fetch (if)**: Read instruction from its memory location into the processor.
- **Instruction operation decoding (iod)**: Analyze instruction to determine type of operation to be performed and operand(s) to be used.
- **Operand address calculation (oac)**: If the operation involves reference to an operand in memory or available via I/O, then determine the address of the operand.
- **Operand fetch (of)**: Fetch the operand from memory or read it in from I/O.
- **Data operation (do)**: Perform the operation indicated in the instruction.
- **Operand store (os)**: Write the result into memory or out to I/O.

3.3 List and briefly define two approaches to dealing with multiple interrupts.

1. Disable all interrupts while an interrupt is being processed.
2. Define priorities for interrupts and to allow an interrupt of higher priority to cause a lower-priority interrupt handler to be interrupted.

3.4 What types of transfers must a computer’s interconnection structure (e.g., bus) support?

- **Memory to processor**: The processor reads an instruction or a unit of data from memory.
- **Processor to memory**: The processor writes a unit of data to memory.
- **I/O to processor**: The processor reads data from an I/O device via an I/O module.
- **Processor to I/O**: The processor sends data to the I/O device.
- **I/O to or from memory**: For these two cases, an I/O module is allowed to exchange data directly with memory, without going through the processor, using direct memory access (DMA).

3.5 What is the benefit of using a multiple-bus architecture compared to a single-bus architecture?

With multiple buses, there are fewer devices per bus. This :

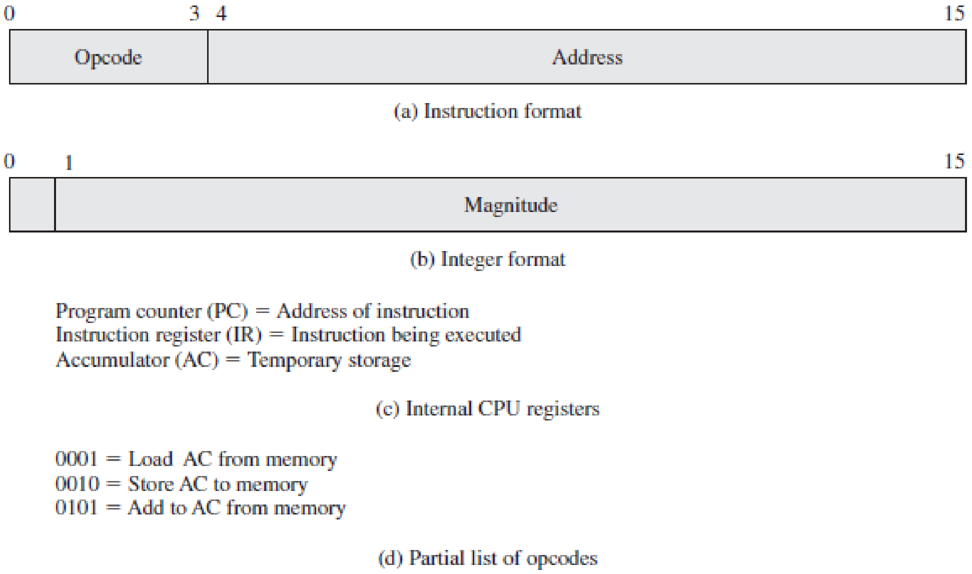
(1) reduces propagation delay, because each bus can be shorter
(2) reduces bottleneck effects.

Problems :

3.1 The hypothetical machine of Figure 3.4 also has two I/O instructions:
0011 = Load AC from I/O
0111 = Store AC to I/O
In these cases, the 12-bit address identifies a particular I/O device. Show the program execution (using the format of figure 3.5) for the following program:

1. Load AC from device 5.
2. Add contents of memory location 0x940.
3. Store AC to device 6.

Assume that the next value retrieved from device 5 is 3 and that location 940 contains a value of 2.



(a) Instruction format

(b) Integer format

Program counter (PC) = Address of instruction
Instruction register (IR) = Instruction being executed
Accumulator (AC) = Temporary storage

(c) Internal CPU registers

0001 = Load AC from memory
0010 = Store AC to memory
0101 = Add to AC from memory

(d) Partial list of opcodes

Figure 3.4 Characteristics of a Hypothetical Machine

Answer :

Instructions OP code : 0001 load Ac from Memory
0010 store Ac to Memory
0101 Add to AC from Memory
0011 Load AC from I/O
0111 store AC to I/O

1. Load AC from device 5. (Load AC from I/O)	Step 1: 3005 → IR; Step 2: 3 → AC
2. Add contents of memory location 0x940.(Add to AC from Memory)	Step 3: 5940 → IR; Step 4: 3 + 2 = 5 → AC
3. Store AC to device 6.(0111 store AC to I/O)	Step 5: 7006 → IR; Step 6: AC → Device 6

3.2 The program execution of Figure 3.5 is described in the text using six steps. Expand this description to show the use of the MAR and MBR.

Answer :

MAR : (memory address register) Address from pc is copied to this in order to be loaded fetched memory
MBR : (memory buffer register) fetched instructions are saved here until they are loaded into IR

1. a. The PC contains 300(Figure 3.4) , the address of the first instruction. This value is loaded in to the MAR.
   b. The value in location 300 (which is the instruction with the value 1940 in hexadecimal) is loaded into the MBR, and the PC is incremented. These two steps can be done in parallel.
   c. The value in the MBR is loaded into the IR.
2. a. The address portion of the IR (940) is loaded into the MAR.
   b. The value in location 940 is loaded into the MBR.
   c. The value in the MBR is loaded into the AC.
3. a. The value in the PC (301) is loaded in to the MAR.

b. The value in location 301 (which is the instruction with the value 5941) is loaded into the MBR, and the PC is incremented.

c. The value in the MBR is loaded into the IR.

4. a. The address portion of the IR (941) is loaded into the MAR.

   b. The value in location 941 is loaded into the MBR.

   c. The old value of the AC and the value of location MBR are added and the result is stored in the AC.

5. a. The value in the PC (302) is loaded in to the MAR.

   b. The value in location 302 (which is the instruction with the value 2941) is loaded into the MBR, and the PC is incremented.

   c. The value in the MBR is loaded into the IR.

6. a. The address portion of the IR (941) is loaded into the MAR.

   b. The value in the AC is loaded into the MBR.

   c. The value in the MBR is stored in location 941.

3.3 Consider a hypothetical 32-bit microprocessor having 32-bit instructions composed of two fields: the first byte contains the opcode and the remainder the immediate operand or an operand address.

a. What is the maximum directly addressable memory capacity (in bytes)?

b. Discuss the impact on the system speed if the microprocessor bus has

1. a 32-bit local address bus and a 16-bit local data bus, or
2. a 16-bit local address bus and a 16-bit local data bus.

c. How many bits are needed for the program counter and the instruction register?

Answer :

a. for this microprocessor , 8 bit for opcode and 24 bit for address so maximum memory addressable is $2^{24} = 16$ MBytes

b.

(1) If the local address bus is 32 bits, the whole address (24 bit) can be transferred at once and decoded in memory. However, because the data bus is only 16 bits, it will require 2 cycles to fetch a 32-bit instruction or operand.

(2) The 16 bits of the address placed on the address bus can't access the whole memory. Thus a more complex memory interface control is needed to latch the first part of the address and then the second part (because the microprocessor will end in two steps to send the whole 24 bits needed).

c. The program counter must be at least 24 bits to be of equal amount as the addresses to be used to access all addresses. The instruction register must be 32 bit to accommodate for the 32 bit instructions

3.4 Consider a hypothetical microprocessor generating a 16-bit address (for example,assume that the program counter and the address registers are 16 bits wide) and havinga 16-bit data bus.

a. What is the maximum memory address space that the processor can access directly if it is connected to a “16-bit memory”?

b. What is the maximum memory address space that the processor can access directly if it is connected to an “8-bit memory”?

c. What architectural features will allow this microprocessor to access a separate “I/O space”?

d. If an input and an output instruction can specify an 8-bit I/O port number, how many 8-bit I/O ports can the microprocessor support? How many 16-bit I/O ports? Explain.

Answer :

16-bit address → $2^{16} = 64$ KB . Memory is byte addressable by default

a–“16-bit memory” means that the maximum accessible data in a single cycle is 16 bits.
so max addressable memory is 64 KB , but 16 bits or 8 bits are accessible in a single cycle.

b–“8-bit memory” means that the maximum accessible data in a single cycle is 8 bits.
so max addressable memory is 64 KB , but only 8bits are accessible in a single cycle.

c- separate input and output instructions are needed, whose execution will generate separate "I/O signals" (different from the "memory signals" generated with the execution of memory-type instructions) .

d- it can support $2^8 = 256$ input and $2^8 = 256$ output byte ports .

3.5 Consider a 32-bit microprocessor, with a 16-bit external data bus, driven by an 8-MHz input clock. Assume that this microprocessor has a bus cycle whose minimum duration equals four input clock cycles. What is the maximum data transfer rate across the bus that this microprocessor can sustain, in bytes/s? To increase its performance, would it be better to make its external data bus 32 bits or to double the external clock frequency supplied to the microprocessor? State any other assumptions you make, and explain. Hint: Determine the number of bytes that can be transferred per bus cycle.

Answer :

Clk cycle time = $\frac{1}{f}$ = $\frac{1}{8*10^6}$ = 125 n sec

bus cycle whose minimum duration equals four input clock cycles :

Bus cycle = 4 * 125 = 500 n sec

16-bit bus :

Maximum data per cycle= 2 Bytes

maximum data transfer rate = Maximum data per cycle/ Bus cycle = $\frac{2}{500*10^{-6}}$

= $4* 10^6$ byte/sec ≈ 3.8 M byte/sec

→If frequency is doubled, memory should be quicker (can work at higher frequencies) to adapt to the change

→if the bus is doubled in size , memory should allow double the current word length (double the data per cycle)

3.7 Consider two microprocessors having 8- and 16-bit-wide external data buses, respectively. The two processors are identical otherwise and their bus cycles take just as long.

a. Suppose all instructions and operands are two bytes long. By what factor do the maximum data transfer rates differ?

b. Repeat assuming that half of the operands and instructions are one byte long.

Answer :

Microprocessors (1) 8-bit-wide external data buses

Microprocessors (2) 16-bit-wide external data buses

a- for 2 byte long instructions :
8-bit can do it in 2 cycles.
16-bit can do it in 1 cycle (has double the transfer rate)

b- for 1 byte long instructions :
8-bit can do it in 1 cycle.
16-bit can do it in 1 cycle (transfer rate advantage is not used)
Example: for 100 transfers, half is 2 byes and half is 1 bytes:
8-bit: number of cycles= 50(1 byte instructions) + 50 * 2 (2 byte instructions) = 150 cycles
16-bit: number of cycles= 50(1 byte instructions) + 50 (2 byte instructions) = 100 cycles