COMPUTER FUNCTION AND INTERCONNECTION

Review questions and solutions.

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

Solution:

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.

Solution:

Instruction address calculation: Determine the address of the next instruction to be executed. Instruction fetch: Read instruction from its memory location into the processor.

Instruction operation decoding: Analyze instruction to determine type of operation to be performed and operand to be used.

Operand address calculation: If the operation involves reference to an operand in memory or available via I/O, then determine the address of the operand.

Operand fetch: Fetch the operand from memory or read it in from I/O.

Data operation: Perform the operation indicated in the instruction.

Operand store: Write the result into memory or out to I/O.

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

Solution:

(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?

Solution:

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).

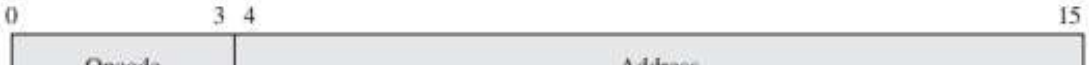
3.5 What is the benefit of using a multiple-bus architecture compared to a single-bus architecture?

Solution:

With multiple buses, there are fewer devices per bus.

(1) reduces propagation delay, because each bus can be shorter, and

(2) reduces bottleneck effects.



(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

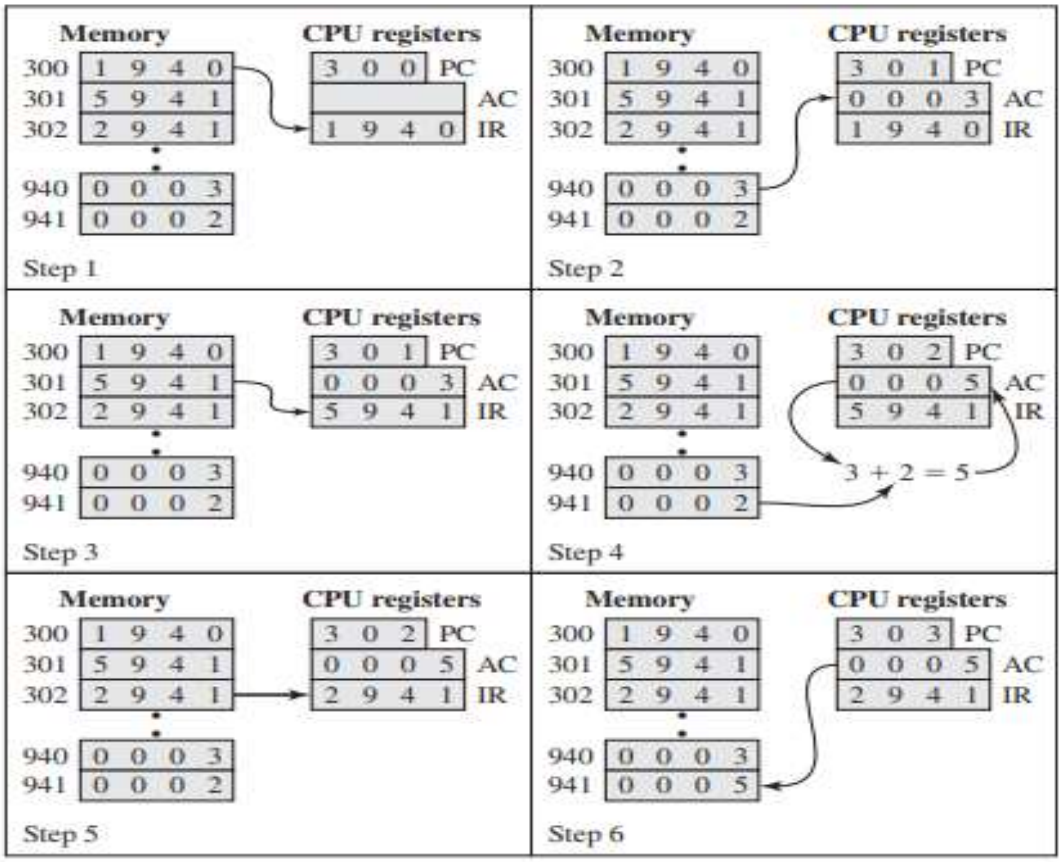


Figure 3.5 Example of Program Execution (contents of memory and registers in hexadecimal)

Problems and solutions

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:

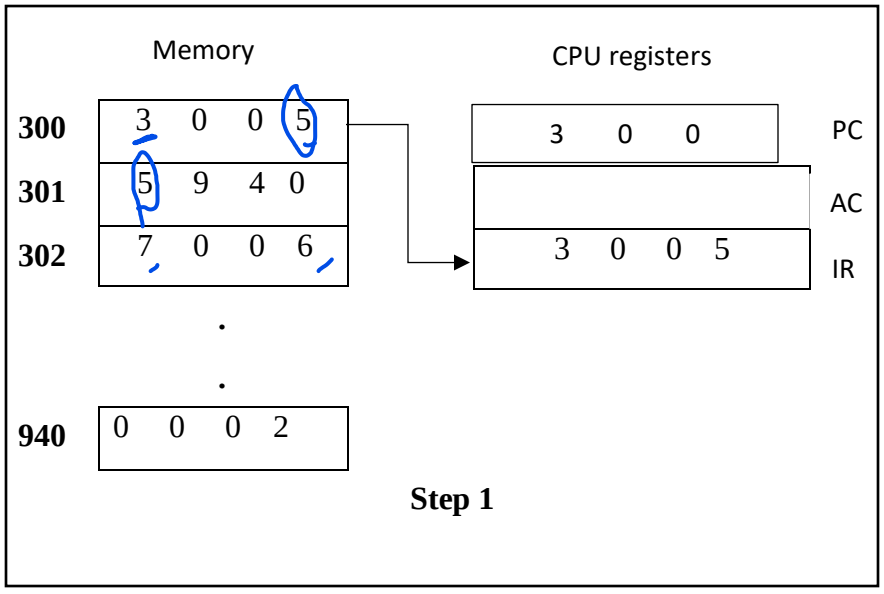
1. Load AC from device 5.

2. Add contents of memory location 940.

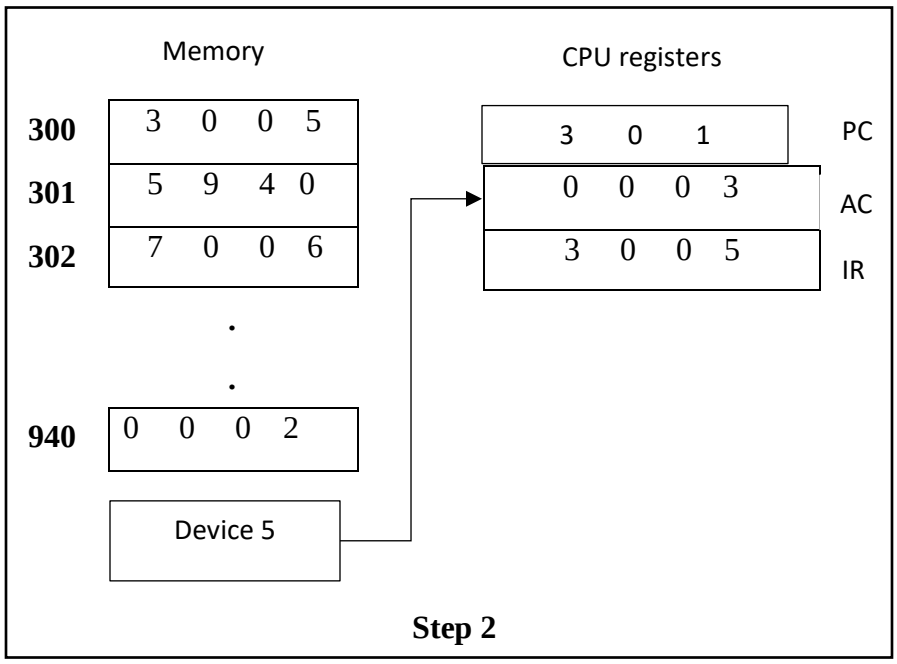
3. Store AC to device 6.

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

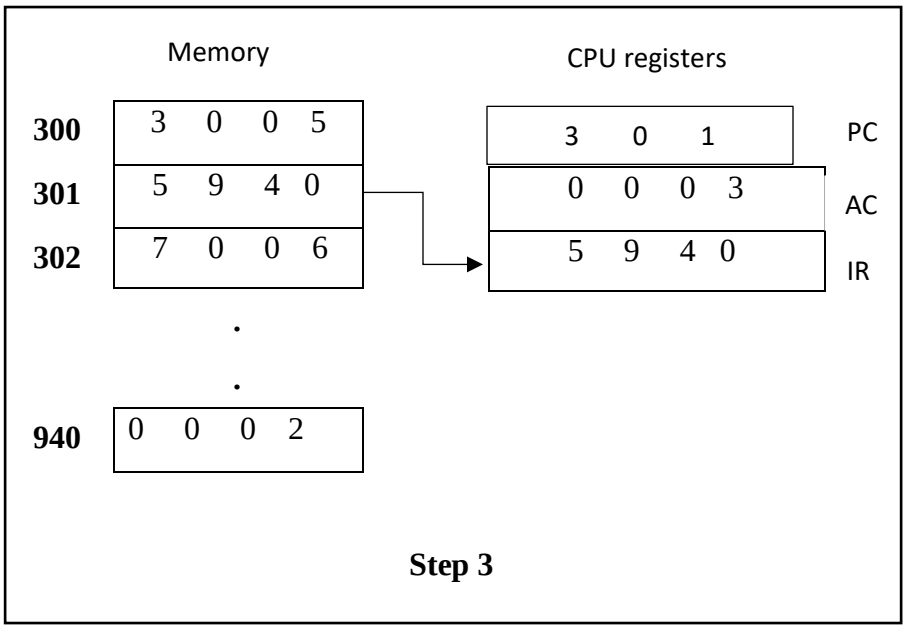
Solution:



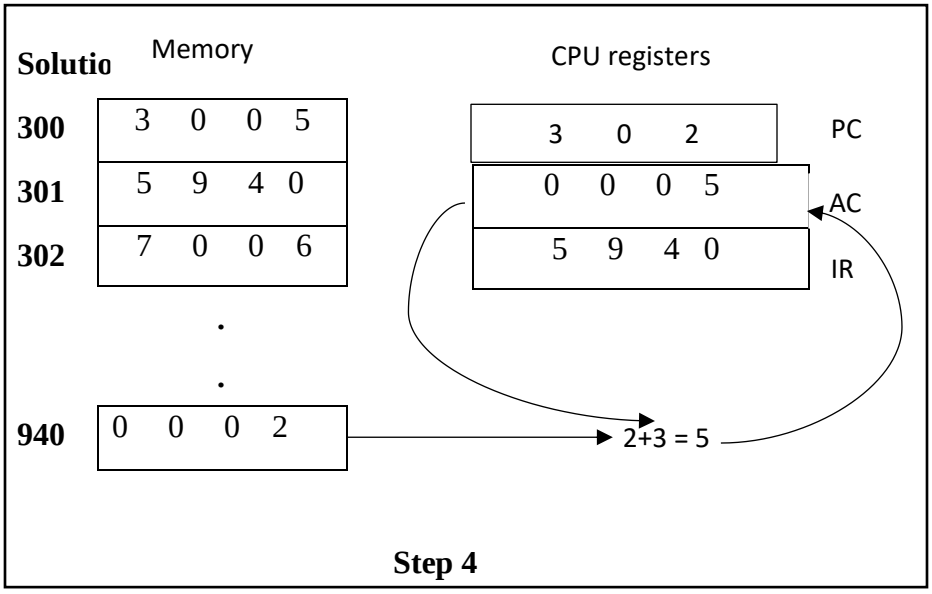
Step 1


Memory
CPU registers
300
3 0 0 5
301
5 9 4 0
302
7 0 0 6
940
0 0 0 2
Device 5
3 0 1
PC
0 0 0 3
AC
3 0 0 5
IR

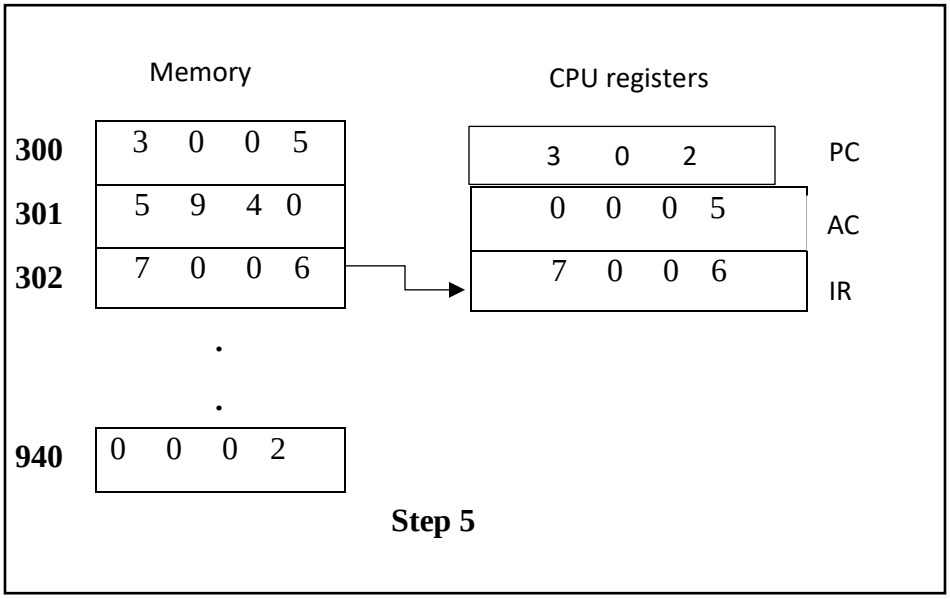
Step 2


Memory
CPU registers
300
3 0 0 5
301
5 9 4 0
302
7 0 0 6
940
0 0 0 2
3 0 1
PC
0 0 0 3
AC
5 9 4 0
IR

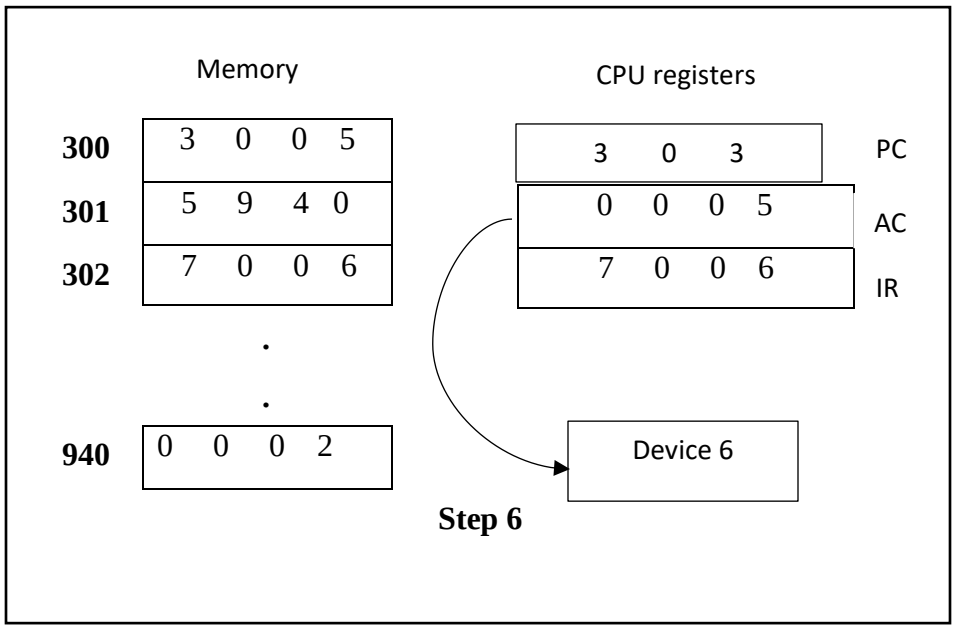
Step 3


Solutio
Memory
CPU registers
300
3 0 0 5
301
5 9 4 0
302
7 0 0 6
940
0 0 0 2
3 0 2
PC
0 0 0 5
AC
5 9 4 0
IR
2+3 = 5

Step 4


Memory
CPU registers
300
3 0 0 5
301
5 9 4 0
302
7 0 0 6
940
0 0 0 2
3 0 2
PC
0 0 0 5
AC
7 0 0 6
IR

Step 5



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.

Solution:

1. a. The PC contains 300, the address of the first instruction. This value is loaded into 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 into 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?

Solution:

a. Since the first byte (8-bits) contains the opcode, so the bits available for operand address = 32-8=24 bit

Thus, the maximum addressable memory, 2^{24} = 16, 777,216 Bytes = 16 MB

b. (1) If the local address bus is 32 bits, the whole address can be transferred at once and decoded in memory. However, because the local 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. In addition, the microprocessor will need 2 cycles to fetch the 32-bit instruction/operand.

c. The program counter must be at least 24 bits (24-bit addresses). If the instruction register is to contain the whole instruction, it will have to be 32-bits long; if it will contain only the opcode (called the opcode register) then it will have to be 8 bits long.

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 having a 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"?

In cases (a) and (b), the microprocessor will be able to access 2^{16} = 64K bytes; the only difference is that with an 8-bit memory each access will transfer a byte, while with a 16-bit memory an access may transfer a word.

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.

Solution:

8 MHz = 1sec
8 000000 clock cycles take 1 sec
1 Clock cycle = 1 /8 MHz = 1/8000000 sec =125 ns
So, 1 Bus cycle takes = 4 × 125 ns = 500 ns (1 bus cycle = 4 input clock cycle)
Given External data bus = 16-bit
Thus, 2 bytes of data can be transferred in every 500 ns
In 500 ns data can be transferred 2 bytes
In 1 sec data can be transferred 4 MB
Transfer rate = 4 MBytes/sec
To increase its performance, if external data bus will be 32 bits then
4 bytes transferred every 500 ns
transfer rate = 8 MBytes/sec
To increase its performance, if double the external clock frequency then
Clock cycle = 1 /16 MHz = 62.5 ns
Bus cycle = 4 × 62.5 ns = 250 ns
2 bytes transferred every 250 ns
transfer rate = 8 MBytes/sec
In the first case, the word length of the memory will have to be double to be able to send or receive 32-bits. In the second case, the speed of the memory chips will also need to double.

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.

Solution:

a. During a single bus cycle, the 8-bit microprocessor transfers one byte while the 16-bit microprocessor transfers two bytes. The 16-bit microprocessor has twice the data transfer rate.

b. Suppose we do 100 transfers of operands and instructions, of which 50 are one byte long and 50 are two bytes long. The 8-bit microprocessor takes 50 + (2 x 50) = 150 bus cycles for the transfer. The 16-bit microprocessor requires 50 + 50 = 100 bus cycles. Thus, the data transfer rates differ by a factor of 150/100=1.5.

3.14 A microprocessor has an increment memory direct instruction, which adds 1 to the value in a memory location. The instruction has five stages: fetch opcode (four bus clock cycles), fetch operand address (three cycles), fetch operand (three cycles), add 1 to operand (three cycles), and store operand (three cycles).

a. By what amount (in percent) will the duration of the instruction increase if we have to insert two bus wait states in each memory read and memory write operation?

b. Repeat assuming that the increment operation takes 13 cycles instead of 3 cycles.

Solution a:

- Without the wait states, the instruction takes 16 bus clock cycles (4+3+3+3+3).
- The instruction requires four memory accesses, resulting in 8 wait states (**fetch opcode: two bus wait states + fetch operand address: two bus wait states + fetch operand: two bus wait states + store operand: two bus wait states)**.
- The instruction, with wait states, takes 24 clock cycles (16+8)
- Duration increased = (Total wait states/Number of cycles without states)*100% = (8/16)*100=50%

b. In this case, the instruction takes 26 bus cycles without wait states and 34 bus cycles with wait states

So, the total wait state =(34-26)=8

Thus, Duration increased = (Total wait states/Number of cycles without wait states) *100%

= (8/26)*100=30.77%

3.17 Consider a 32-bit microprocessor whose bus cycle is the same duration as that of a 16-bit microprocessor. Assume that, on average, 20% of the operands and instructions are 32 bits long, 40% are 16 bits long, and 40% are only 8 bits long. Calculate the improvement achieved when fetching instructions and operands with the 32-bit microprocessor.

Solution:

- Consider a mix of 100 instructions and operands.
- The number of bus cycles required for the 16-bit microprocessor is (2 × 20) + 40 + 40 = 120.
- For the 32-bit microprocessor, the number required is 20+ 40 + 40 =100.
- This amounts to an improvement of (120-100)*100/120 = 16.67%.