**State the function of BHE and A0 pins of 8086**

1. BHE:

* Stands for Bus High Enable. It is available at pin 34 and used to indicate the transfer of data using data bus D8-D15. This signal is low during the first clock cycle, thereafter it is active.

1. A0 (Address 0) Pin of 8086:

* The A0 pin differentiates between memory and I/O operations, allowing the processor to direct the data transfer to the appropriate memory or I/O device.

|  |  |  |
| --- | --- | --- |
| **BHE** | **A0** | **Operation Performed** |
| 0 | 0 | Whole word form even address |
| 0 | 1 | Upper Byte to / to even address |
| 1 | 0 | Lower Byte to / to odd address |
| 1 | 1 | None |

**How single stepping or tracing is implemented in 8086?**

If the trap flag is set, the 8086 will automatically do a type-1 interrupt after each instruction executes. When the 8086 does a type-1 interrupt, it pushes the flag register on the stack.

**Explain memory segmentation in 8086 and list its advantages. (any two)**

1. Memory segmentation in the 8086 microprocessor refers to the division of the memory into multiple segments, each with a specific starting address and length. The 8086 uses a segmented memory model where the physical memory is organized into 64KB segments.
2. Code segment register (CS): is used for addressing memory location in the code segment of the memory, where the executable program is stored.
3. Data segment register (DS): points to the data segment of the memory where the data is stored.
4. Extra Segment Register (ES): also refers to a segment in the memory which is another data segment in the memory.
5. Stack Segment Register (SS): is used for addressing stack segment of the memory. The stack segment is that segment of memory which is used to store stack data.

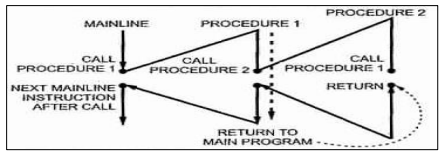
**Advantages**

* It provides a powerful memory management mechanism.
* Data related or stack related operations can be performed in different segments.
* Code related operation can be done in separate code segments.
* It allows to processes to easily share data.

**With neat sketches demonstrate the use of re-entrant and recursive procedure.**

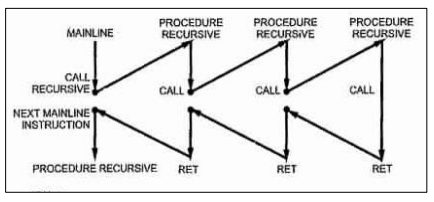
1. **Reentrant Procedure:**

* A reentrant procedure is one in which a single copy of the program code can be shared by multiple users during the same period of time. Re-entrance has two key aspects: The program code cannot modify itself and the local data for each user must be stored separately.



1. **Recursive procedures:**

* An active procedure that is invoked from within itself or from within another active procedure is a recursive procedure. Such an invocation is called recursion. A procedure that is invoked recursively must have the RECURSIVE attribute specified in the PROCEDURE statement.



**1. Immediate addressing mode:**

An instruction in which 8-bit or 16-bit operand (data) is specified in the instruction, then the addressing mode of such instruction is known as Immediate addressing mode.

Example:

MOV AX,67D3H

**2. Register addressing mode**

An instruction in which an operand (data) is specified in general purpose registers, then the addressing mode is known as register addressing mode.

Example:

MOV AX,CX

**3. Direct addressing mode**

An instruction in which 16 bit effective address of an operand is specified in the instruction, then the addressing mode of such instruction is known as direct addressing mode.

Example:

MOV CL,[2000H]

**4. Register Indirect addressing mode**

An instruction in which address of an operand is specified in pointer register or in index register or in BX, then the addressing mode is known as register indirect addressing mode.

Example:

MOV AX, [BX]

**5. Indexed addressing mode**

An instruction in which the offset address of an operand is stored in index registers (SI or DI) then the addressing mode of such instruction is known as indexed addressing mode.

DS is the default segment for SI and DI. For string instructions DS and ES are the default segments for SI and DI resp. this is a special case of register indirect addressing mode.

Example:

MOV AX,[SI]

**6. Based Indexed addressing mode:**

An instruction in which the address of an operand is obtained by adding the content of base register (BX or BP) to the content of an index register (SI or DI) The default segment register may be DS or ES

Example:

MOV AX, [BX][SI]

**7. Register relative addressing mode:**

An instruction in which the address of the operand is obtained by adding the displacement (8-bit or 16 bit) with the contents of base registers or index registers (BX, BP, SI, DI). The default segment register is DS or ES.

Example:

MOV AX, 50H[BX]

**8. Relative Based Indexed addressing mode**

An instruction in which the address of the operand is obtained by adding the displacement (8 bit or 16 bit) with the base registers (BX or BP) and index registers (SI or DI) to the default segment.

Example:

MOV AX, 50H [BX][SI]