MP FA 2 ANS

1. Explain protected mode address translation mechanism to convert 48 bits logical address to 32 bits Linear address.

   Ans:

Protected mode is a mode in modern computer processors that provides hardware-based memory protection for operating systems. In this mode, the processor uses a mechanism called paging to translate logical addresses used by programs into physical addresses in computer memory. The process involves several steps to convert a 48-bit logical address to a 32-bit linear address. Here's how it works:

1. **Segmentation:** In protected mode, memory is divided into segments, and each segment has a base address and a limit. The logical address consists of two parts: the segment selector (16 bits) and the offset within the segment (32 bits). The segment selector is an index into a descriptor table that contains information about the segment, including its base address.
2. **Descriptor Table:** The descriptor table holds segment descriptors. A segment descriptor contains information such as the base address of the segment, the segment's limit (size), and access control information. When a segment selector is provided, the processor uses it to locate the corresponding segment descriptor in the descriptor table.
3. **Segment Base Address:** The base address in the segment descriptor represents the starting address of the segment in physical memory.
4. **Logical Address Calculation:** To calculate the logical address, the offset within the segment is added to the base address obtained from the segment descriptor. This calculation results in a 48-bit logical address.
5. **Paging:** Once the logical address is calculated, the paging mechanism comes into play. Paging divides physical memory into fixed-size blocks called pages. The processor uses a page table to translate the page number in the logical address to a frame number in physical memory. Each entry in the page table contains the base address of the corresponding page in physical memory.
6. **Linear Address:** The page table lookup results in a 32-bit frame number. This frame number, combined with the offset within the page from the logical address, forms the 32-bit linear address. The linear address represents the actual location in physical memory where the data is stored.

By using segmentation and paging together, protected mode provides a flexible and secure way of managing memory, allowing modern operating systems to run multiple processes simultaneously without interfering with each other's memory spaces.

2. List the application of control registers in 80386 DX processor? Outline structure of control registers and explain the function of all bits.

Ans: The 80386 DX processor, part of the Intel x86 family of microprocessors, uses control registers to manage various aspects of its operation. These control registers play a crucial role in tasks such as memory management, task switching, and system control. Here is an overview of the structure of control registers in the 80386 DX processor along with the functions of their bits:

Control Registers in 80386 DX Processor:

1. **CR0 (Control Register 0):**
   - **Bit 0 (PE):** Protection Enable. When set, enables protected mode operation.
   - **Bit 1 (MP):** Monitor Coprocessor. Controls interaction of WAIT/FWAIT instructions with the numeric coprocessor.
   - **Bit 2 (EM):** Emulation. If set, no x87 floating-point instructions are executed; they trigger exceptions.
   - **Bit 3 (TS):** Task Switched. Indicates if a task switch has occurred since last reset.
   - **Bit 4 (ET):** Extension Type. If set, the external math coprocessor is present.
   - **Bits 16-31:** Reserved.
2. **CR1 (Control Register 1):**
   - Not used in 80386 DX processor.
3. **CR2 (Control Register 2):**
   - Contains the physical address of the page directory used in paging operations.
4. **CR3 (Control Register 3):**
   - Contains the physical address of the page directory used in virtual memory translation.
5. **CR4 (Control Register 4):**
   - **Bit 0 (VME):** Virtual 8086 Mode Extensions. Enables support for the virtual interrupt flag (VIF) and virtual interrupt pending flag (VIP).
   - **Bit 1 (PVI):** Protected-mode Virtual Interrupts. Enables hardware support for virtual interrupts.
   - **Bit 2 (TSD):** Time Stamp Disable. Disables the RDTSC instruction for non-supervisor mode programs.
   - **Bit 3 (DE):** Debugging Extensions. Enables I/O breakpoints and extra debug registers.
   - **Bit 4 (PSE):** Page Size Extensions. Enables 4 MB pages.
   - **Bit 5 (PAE):** Physical Address Extensions. Enables physical addresses greater than 32 bits.
   - **Bit 6 (MCE):** Machine Check Exception. Enables machine check interrupts.
   - **Bit 7 (PGE):** Page Global Enable. Enables global pages.
   - **Bit 8 (PCE):** Performance-Monitoring Counter Enable.
   - **Bit 9 (OSFXSR):** OS Support for FXSAVE and FXRSTOR Instructions.
   - **Bit 10 (OSXMMEXCPT):** OS Support for Unmasked SIMD Floating-Point Exceptions.
   - **Bit 11-31:** Reserved.

These control registers provide essential control and configuration options for the 80386 DX processor, allowing it to handle different operating modes, memory management, and system control tasks effectively. Understanding and appropriately configuring these registers are vital for system stability and performance.

3. Explain the problems occurring in pipeline processor due to branch instructions. How the flushing of the instruction pipeline can be minimized in Pentium processor?

Ans: In pipelined processors, branch instructions can cause several problems due to their conditional nature. When a branch instruction is encountered, the processor must decide whether to take the branch (change the program counter to a new address) or continue with the next instruction in sequence. Here are the issues related to branch instructions in pipeline processors:

1. **Pipeline Stalls:** If the branch instruction is a conditional branch, the decision to take the branch or not is made late in the pipeline stages. Until the branch condition is evaluated, subsequent instructions cannot proceed, causing pipeline stalls. Stalls reduce the performance gain achieved by pipelining, as stages sit idle waiting for the branch decision.
2. **Mis predicted Branches:** If the branch prediction logic predicts the branch direction incorrectly, the pipeline is filled with instructions that shouldn't have been executed, leading to wasted computation. When the misprediction is detected, the pipeline must be flushed, and correct instructions must be fetched and executed. This is a significant performance penalty.

To minimize the flushing of the instruction pipeline in the Pentium processor and handle branch instructions more efficiently, several techniques are employed:

1. **Branch Prediction:** Modern processors, including the Pentium, use sophisticated branch prediction algorithms to predict the outcome of branch instructions. These predictions are based on historical behavior and patterns. If the prediction is correct, the pipeline continues to execute instructions without stalls. If incorrect, the pipeline is flushed, but this happens less frequently due to accurate predictions.
2. **Speculative Execution:** Pentium processors use speculative execution to continue executing instructions past a branch instruction without knowing whether the branch will be taken or not. Speculatively executed instructions are later discarded if the branch prediction was incorrect. This technique helps in keeping the pipeline busy and improving overall throughput.
3. **Out-of-Order Execution:** Pentium processors employ out-of-order execution, allowing instructions that are not dependent on the branch instruction to continue execution. This helps in utilizing the pipeline resources effectively even when a branch instruction is in the pipeline.
4. **Branch Target Buffer (BTB):** The BTB is a cache that stores branch target addresses and their corresponding outcomes (taken or not taken). It helps in quickly fetching the correct target address of branch instructions, reducing the pipeline stall time in case of a mis predicted branch.

By using these techniques, modern processors like the Pentium minimize the impact of branch instructions on the pipeline, ensuring better performance and efficient utilization of resources.

Q.4 Explain in brief, integer pipeline stages of Pentium processor. Also show super pipeline operation of Pentium.

Ans: The Pentium processor, introduced by Intel, employs a pipelined architecture to enhance its performance. In a pipelined processor, the instruction execution is divided into stages, allowing multiple instructions to be processed simultaneously at different stages of the pipeline. The Pentium processor has five integer pipeline stages:

Integer Pipeline Stages of Pentium Processor:

1. **Fetch (F):** In this stage, instructions are fetched from memory. The instruction pointer is used to fetch the next instruction from the memory address pointed to by the instruction pointer.
2. **Decode (D):** The fetched instruction is decoded in this stage. Opcode and operand information are extracted, and the necessary control signals are generated for subsequent stages.
3. **Execute (E):** In this stage, arithmetic and logic operations are performed. ALU (Arithmetic Logic Unit) operations, such as addition, subtraction, and bitwise operations, take place here. Memory address calculations for data access also occur in this stage.
4. **Memory (M):** Memory operations, such as reading from or writing to memory, are executed in this stage. Load and store operations are performed here.
5. **Write-Back (W):** The results of the executed instructions are written back to the appropriate register in this stage. This stage updates the register file with the results of ALU operations or data loaded from memory.

Super Pipeline Operation of Pentium:

In addition to the five basic stages, the Pentium processor also incorporates a concept known as super pipelining. Super pipelining involves breaking down the basic pipeline stages further into smaller sub-stages, allowing even finer instruction-level parallelism. While the exact sub-stages and their number can vary, here's a simplified overview:

- **Fetch1 (F1):** Initial fetch stage where the instruction is fetched from memory.
- **Fetch2 (F2):** Further processing of the fetched instruction, preparing it for execution.
- **Decode1 (D1):** Partial decoding of the instruction.
- **Decode2 (D2):** Complete decoding, generating necessary control signals.
- **Execute1 (E1):** Initial execution stage where some basic operations begin.
- **Execute2 (E2):** Continued execution, possibly involving more complex ALU operations.
- **Memory1 (M1):** Initial memory access, preparing for read/write operations.
- **Memory2 (M2):** Completion of memory operations.
- **Write-Back (W):** Writing the results back to the register file.

By breaking down the stages into sub-stages, the Pentium processor can overlap the execution of multiple instructions to a greater extent, improving its overall throughput and performance. Super pipelining allows the processor to handle a larger number of instructions simultaneously, making it more efficient in executing complex instruction sequences. Keep in mind that the actual architecture and sub-stage division can vary between different Pentium processor models.

5) Write the real time application of 8051 microcontrollers. Explain the features of 8051 microcontrollers.

Ans: **Real-Time Applications of 8051 Microcontrollers:**

8051 microcontrollers find applications in various real-time systems due to their versatility, reliability, and ease of use. Some common real-time applications include:

1. **Embedded Systems:** 8051 microcontrollers are widely used in embedded systems such as home appliances (microwaves, washing machines), industrial control systems, and automotive control systems.
2. **Automated Systems:** 8051 microcontrollers are used in automated systems like traffic light control, automatic toll collection systems, and industrial automation.
3. **Robotics:** 8051 microcontrollers serve as the brain of many robots, controlling their movements, sensors, and actuators.
4. **Medical Devices:** They are employed in various medical devices like blood pressure monitors, insulin pumps, and digital thermometers.
5. **Instrumentation:** 8051 microcontrollers are used in instruments like oscilloscopes, function generators, and multimeters for data processing and control.
6. **Consumer Electronics:** They find applications in consumer electronics such as TV remotes, air conditioners, and home entertainment systems.

Features of 8051 Microcontrollers:

8051 microcontrollers have several features that make them suitable for a wide range of applications:

1. **CPU Core:** The 8051 microcontroller has an 8-bit CPU core that can execute a wide variety of instructions.
2. **Memory:** It typically has on-chip RAM for data storage and on-chip Flash memory for program storage. Some variants also support external memory.
3. **I/O Ports:** 8051 microcontrollers come with multiple I/O ports that can be used to interface with external devices and sensors.
4. **Timers/Counters:** They have built-in timers/counters that are useful for generating precise time delays and generating PWM signals.
5. **Serial Communication:** 8051 microcontrollers support serial communication protocols like UART (Universal Asynchronous Receiver/Transmitter), which is crucial for communication between microcontrollers and other devices.
6. **Interrupt System:** They have an interrupt system to handle external events asynchronously, allowing the microcontroller to respond to real-time events promptly.

7. **Low Power Modes:** 8051 microcontrollers often have low-power modes, making them suitable for battery-operated and energy-efficient applications.
8. **Clock Circuit:** They contain an on-chip oscillator or can be interfaced with an external crystal oscillator for generating clock pulses.
9. **Analog-to-Digital Converter (ADC):** Some variants have an in-built ADC, which allows the microcontroller to convert analogy signals from sensors to digital values.
10. **Instruction Set:** The 8051 microcontroller has a rich set of instructions, enabling it to perform a wide range of operations.

These features make 8051 microcontrollers popular in various real-time applications, especially those that require precise control, communication, and interfacing capabilities.