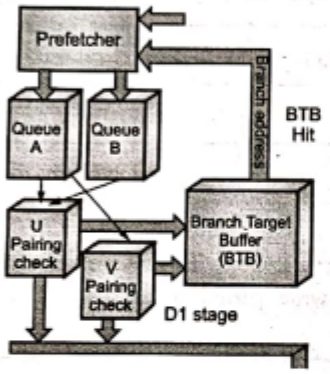
Q1.
https://www.ques10.com/p/10876/explain-the-interrupt-structure-of-8086-processo-1/
Q2.
https://www.tutorialspoint.com/microprocessor/microprocessor_8257_dma_controller.htm
Q3.

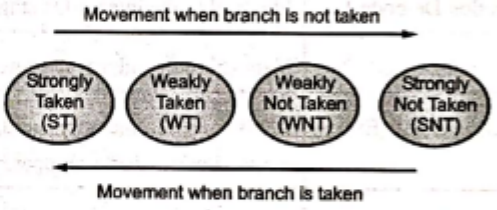
✓Syllabus Topic : Branch Prediction Logic

11.5 Pentium I Branch Prediction Logic

Q. Explain branch prediction logic used in Pentium. (Dec. 15, 10 Marks)
Q. Explain, in brief, branch prediction mechanism is on Pentium processor. (Dec. 16, 10 Marks)
Q. Write short note on : Branch prediction logic. (May 17, 5 Marks)
Q. How flushing problem is minimized in Pentium? Explain. (Dec. 17, 10 Marks)
Q. Explain the branch prediction logic used in Pentium processor (Dec. 18, 10 Marks)



(1G2)Fig. 11.5.1 : Branch Prediction Logic



(1G3)Fig. 11.5.2 : History bits

- The Pentium processor includes branch prediction logic, allowing it to minimize pipeline flushing.
- When a branch operation is correctly predicted, no performance penalty is incurred.
- However, when branch prediction is not correct, a three cycle penalty is incurred if the branch is executed in the U pipeline and upto four (3+1 extra may be needed) cycle penalty if the branch is in the V pipeline.
- The prediction mechanism is implemented using a four-way, set-associative Cache with 256 entries.
- This is referred to as the **Branch Target Buffer**, or BTB.
- The directory entry for each line contains the following information.
  1. **A valid bit** that indicates whether or not the entry is in use.
  2. **Two History bits** that-track how often the branch has been taken each time that it entered the pipeline before.
  3. The **Memory Address** of the branch instruction for identification.
- The **Branch Target Buffer**, or BTB, is a look-aside cache that sits off to the side of the Dl stages of the two pipelines and monitors for branch instructions.
- During D1 stage, when an instruction is decoded and identified as a branch instruction, the address of the instruction is searched in the BTB for a previous history.

- If no history exists, then prediction is made that the branch will not be taken.
- If there is a history (BTB hit), then prediction is made as follows:
  - If the History bits are 00 or 01 (Strongly Not taken or weakly not taken), then the prediction is that the branch will not be taken.
  - If the History bits are 10 or 11 (Strongly taken or weakly taken), then the prediction is that the branch will be taken.
- **If the branch is predicted to be taken**, then the active queue is no longer used. Instead, the prefetcher starts fetching instructions from the branch address and stores them into the second queue which now becomes the active queue. This queue now starts feeding instructions into the two pipes.
- **If branch is predicted to be not taken**, then nothing changes, and the active queue remains active and instructions are fetched from the sequentially next locations.
- When the instruction reaches the execution stage, the branch will either be taken or not taken. If taken, the next instruction to be executed should be the one fetched from the branch target address,
- If the branch is not taken the next instruction executed should be the one fetched from the next sequential memory address after the branch instruction.
- When the branch is taken for the first time, the execution unit provides feedback to the prediction logic. The branch target address is sent back and recorded in the BTB.
- A directory entry is made containing the source memory address that the branch instruction was fetched from and history bits are set to .indicate that the branch has been strongly taken.
- The history bits can indicate one of four possible states.

History Bits	Resulting Description	Prediction made	If actually taken	If actually not taken
00	Strongly Not Taken	Branch Will Not Be Taken	Upgrades to Weakly Not Taken	Remains Strongly Not Taken
01	Weakly Not Taken	Branch Will Not Be Taken	Upgrades to Weakly Taken	Downgrades to Strongly Not Taken
10	Weakly Taken	Branch Will Be Taken	Upgrades to Strongly Taken	Downgrades to Weakly Not Taken
11	Strongly Taken	Branch Will Be Taken	Remains Strongly Taken	Downgrades to Weakly Taken

During execution stage the μP realizes whether the prediction was true or false. The following actions thus take place….

1. If the branch was correctly predicted taken, the entry's history bits are upgraded and no further action is necessary. The correct instructions are already in the pipelines behind the branch instruction.
2. If the branch was incorrectly predicted taken, the entry is downgraded. The instructions in the pipelines behind the branch are incorrect and must be flushed. The branch prediction logic instructs the prefetcher to switch back to the other queue and resume sequential code fetches.
3. If the branch was correctly predicted not taken and there is a corresponding entry in the BTB, downgrade the entry's history bits. If the branch was correctly predicted not taken (because a BTB miss occurred in the D1 stage) and there isn't a corresponding entry in the BTB, do not make an entry in the BTB.
4. If the branch was incorrectly predicted not taken and there is a corresponding entry in the BTB, upgrade the entry's history bits. If the branch was incorrectly predicted not taken and there isn't a corresponding entry in the BTB, make an entry and mark it strongly taken.

Q4.
https://www.ques10.com/p/13578/compare-8086-80386-and-pentium-1/
Q5.
https://mmrcse.blogspot.com/2018/12/draw-and-explain-internal-architecture.html
Q7.

https://www.ques10.com/p/10910/differentiate-between-minimum-and-maximum-mode-of-/?
https://www.geeksforgeeks.org/architecture-of-8086/#:~:text=8086%20provides%20the%20programmer%20with,it%20into%2016%2064kB%20segments.

Q10.
Nahi mila
Q11.
https://www.geeksforgeeks.org/memory-banking-in-microprocessor/#:~:text=The%208086%20processor%20provides%20a,this%20problem%20is%20Memory%20Banking.

Q12.
https://www.tutorialspoint.com/8255-microprocessor-operating-modes#:~:text=These%20are%20used%20to%20set,11%2C%20it%20is%20m2%20mode.&text=1%20when%20port%20A%20is,port%20A%20is%20sending%20output.&text=1%20when%20higher%20nibble%20of,higher%20nibble%20is%20sending%20output.
Q13.
https://www.tutorialspoint.com/microprocessor/microprocessor_8086_interrupts.htm#:~:text=Interrupt%20is%20the%20method%20of,how%20to%20handle%20the%20interrupt.
Q14.
https://eduladder.com/viewquestions/10396/Explain-integer-pipeline-of-Pentium-processor
Q15.

Hyper-threading Technology

Strategies of Implementation

There are several strategies to implement hyperthreading.

As has been mentioned earlier, a single processor can execute multiple threads by switching between them. The scheme of context switching may again be of several types.

They are:

(i) *Time-slice multithreading* In this scheme the processor switches between different process threads after a fixed time slice. Since there is only one processor, Time-slice multithreading can result in loss of execution cycles. However, each thread is guaranteed to get attention of the processor when its turn comes. In case there is a cache miss, which is a long latency event, automatically the processor will switch to another thread.

(ii) *On chip multiprocessing* (CMP) This scheme involves the use of two processors on a single die. The two processors each have a full set of execution and architectural resources. The processors may or may not share a large on-chip cache. CMP is largely orthogonal to conventional multi-processor systems, as you can have multiple CMP processors in a multiprocessor configuration. CMP chip is significantly larger than the size of single core-chip and therefore more expensive to manufacture (i) the die size is obviously more and (ii) power consumption is also higher.

(iii) *Hyperthreading* Finally, there is simultaneous multi-threading, where multiple threads can be executed on a single processor without switching. The threads execute simultaneously and make much better use of the resources. This approach makes the most effective use of processor resources. How can it be done ?

This new technology which was first introduced on the Intel Xeon processor in early 2002 was subsequently employed in Pentium 4 in November 2002 at clock frequencies of 3.06 GHz and higher.

In Pentium architecture a single physical processor appears as two logical processors. All the physical resources of the system are shared between these two logical processors. This means that the user programs can schedule the processes or threads to the two logical processors as if there are indeed two different physical processors. The instructions from both the logical processors can be executed simultaneously on shared execution resources.

Hyperthreading used the concept of simultaneous multithreading and shows an improvement in the Intel microarchitecture development. At the expense of an added cost of less than only 5 percent in added die area, the performance increases by about 25 percent.

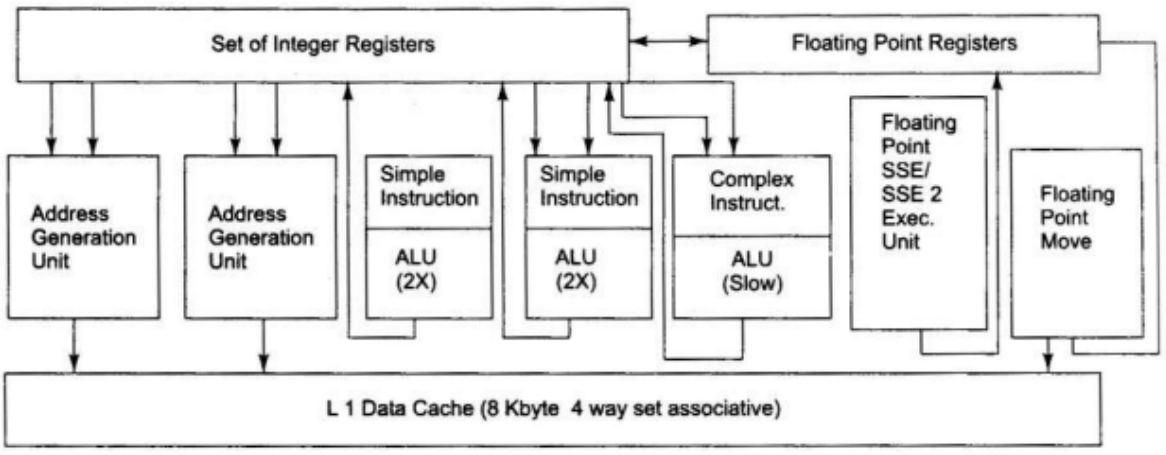
The major elegance of this architecture lies in devising appropriate resource sharing policy for each shared resource. Several resource sharing strategies have been investigated by the developers. Some of these are (a) partitioned resources, (b) threshold sharing, and (c) full sharing. The choice of sharing strategy to be adopted depends on several factors, such as, the traffic pattern, size of the resource, potential deadlock probabilities and other considerations.

To do this, there is one copy of the architecture state for each logical processor, and the logical processors share a single set of physical execution resources. From a software or architecture perspective, this means operating systems and user programs can schedule processes or threads to

Hyper-threading Technology

logical processors as they would on conventioal physical processors in a multi-processor system. From a microarchitecture perspective, this means that instructions from logical processors will persist and execute simultaneously on shared execution resouces.

Figure shows a multiprocessor system with two processors which does not incorporate Hyperthreading Technology.



Q18.

80386: Protected Mode

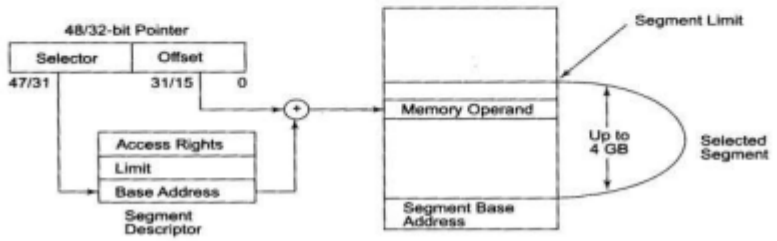
All the capabilities of 80386 are available for utilization in its protected mode of operation. In this mode, the 80386 can address 4 Gigabytes of physical memory and 64 terrabytes of virtual memory per task. The 80386 in the protected mode supports all softwares written for 80286 and 8086 to be executed under the control of memory management and protection abilities of 80386. The protected mode allows the use of additional instructions, addressing modes and capabilities of 80386.

Addressing in Protected Mode

80386 support two methods
a) Paging disabled b) Paging enabled

a) Paging disabled

In this mode, the contents of segment registers are used as selectors to address descriptors which contain the segment limit, base address and access rights byte of the segment. The effective address (offset) is added with segment base address to calculate linear address. This linear address is further used as physical address, if the paging unit is disabled. Otherwise, the paging unit converts the linear address into physical address.



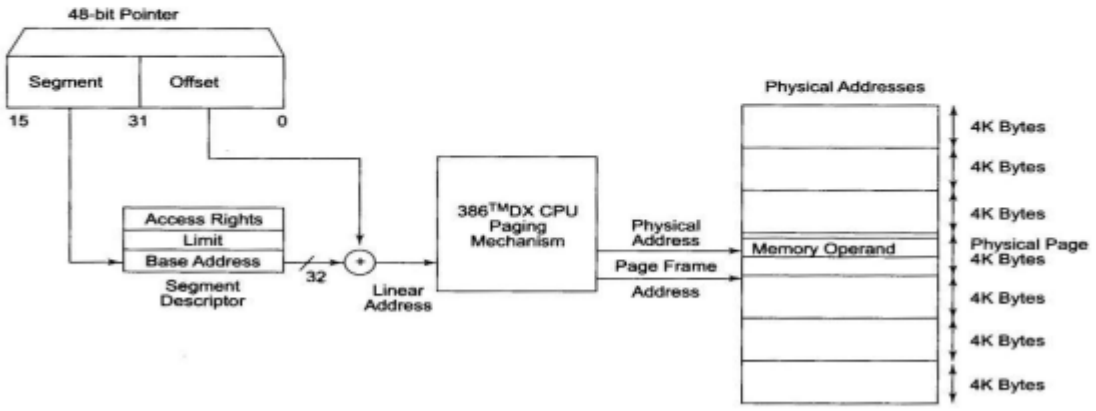
Protected Mode Addressing without Paging Unit (Intel Crop.)

23

80386: Protected Mode

b) Paging Enabled

The paging unit is a memory management unit enabled only in the protected mode. The paging mechanism allows handling of large segments of memory in terms of pages of 4 Kbyte size. The paging unit operates under the control of segmentation unit. The paging unit if enabled converts linear addresses into physical addresses, in protected mode.



Paging Unit Enabled in Protected Mode Addressing (Intel Corp.)

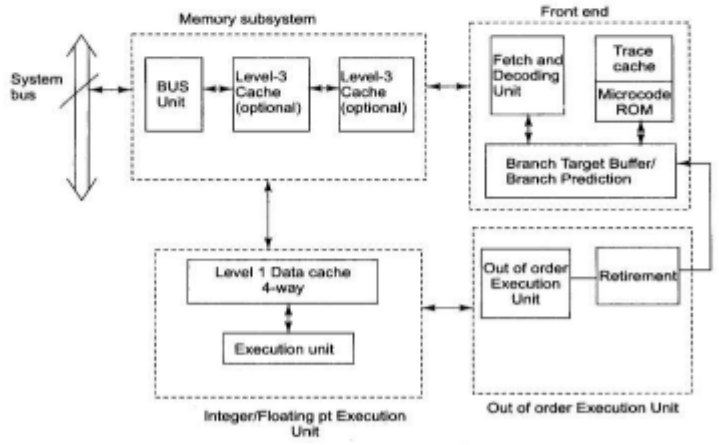
Q19.
https://www.geeksforgeeks.org/memory-segmentation-8086-microprocessor/
Q20.
https://www.ques10.com/p/13378/draw-and-explain-timing-diagram-for-read-operati-1/
Q21.
https://www.ques10.com/p/13317/explain-programming-model-of-8086-1/#:~:text=The%20programming%20model%20of%20the,programming%20model%20of%208086%20microprocessor.
Q22.
https://www.includehelp.com/embedded-system/the-bsr-mode-of-8255-ppi-programmable-peripheral-interface.aspx
Q23.

Pentium 4: Net burst micro architecture

- A fast processor requires balancing and tuning of many microarchitectural features that compete for processor die cost and for design and validation efforts.
- In Intel's Net Burst microarchitecture of the Pentium 4 processor, there are four main sections:
  - the in-order front end,
  - the out-of-order execution engine,
  - the integer and floating-point execution units, and
  - the memory subsystem.

Net burst Micro Architecture for Pentium – 4

The pentium 4 architecture may be viewed as having four basic modules, (A) Front end module (B) Out of order execution engine (C) Execution module and (D) Memory subsystem module. This has been shown in Fig.



Block Diagram of Pentium 4 Microarchitecture

Front End Module:

Front End Module:

- Function of unit is to fetch the instructions to be executed ,decoded them and pass these decoded instructions to the out of order execution module.

Microcode ROM:

- Complex instructions such as interrupts or string manipulations are decoded in microcode ROM.Once decoding is done these are passed back to trace cache and then fed to execution engine.

Decode Unit:

- Instruction decoder decodes instructions and translate them into micro operations.it decodes one instruction per clock cycle.
- Simple instructions are translated into single micro-op while complex ones are translated into multiple number of micro-ops.

Branch Prediction:

- This unit tells the locations from where the next instruction bytes are fetched.
- Predictions are based on past history of the program execution.
- There are 2 types of branch predictions: 1.Static 2.Dynamic.

Out of Order Execution Engine:

- It consists of allocation ,Register renaming ,scheduling and execution functions.

Trace Cache:

- Trace Cache is a special instruction cache in which stream of instructions(Micro-operations) are not stored but decoded and fed to L1 cache. This enhances execution speed.
- It can store up to 12K micro-ops. This cache assembles the decoded micro-ops into ordered sequence called traces, hence named as trace cache.

Q25.
https://electronicsdesk.com/80386-microprocessor.html