What's New in Computers

Intel's New P6 Processor

Vijnan Shastri

Vijnan Shastri works at the Centre for Electronics Design Technology in the Indian Institute of Science, Bangalore , and his areas of interest are digital and microprocessor systems, storage sub-systems and systems software.

This article briefly describes the architecture of a new microprocessor, called P6, being released by Intel Corporation.

The P6 is the next processor to be released (sometime in 1996) by Intel after the Pentium—currently the latest processor from them. P6 incorporates many advanced features such as superpipelining, super-scalar architecture, branch-prediction and advanced caching. We will explore these features (see page 99) in the following paragraphs and learn what these terminologies mean.

Super-Pipelining and Super-Scalar Architecture

A state machine is a digital system driven by a clock that goes through various states (referred to as state transitions) on every ' tick ' (clock cycle).

Any processor is in fact a complex state machine. A state machine is a digital system driven by a clock that goes through various states (referred to as state transitions) on every 'tick' (clock cycle). The transitions through states depend on two sets of inputs: the current state and the outside world inputs. The outputs of the state machine are decoded from (obtained from) the states of the machine. For instance, if you were to build a digital stop watch, then you would build a state machine with a clock period of one second and the inputs would be the current time, start, stop and reset. The most important input for a processor is the set of instructions it fetches. The main steps involved in this process are *instruction fetch, decode* and *execute*. Each of these operations takes some time. In early processors such as 8085, these operations took place serially because the state machine was designed as a single monolithic unit. However, in current microprocessors, the state machine is divided into independent units (which interact with one another). This means that the instruction fetch unit fetches an instruction and hands it over to the decoding unit. While the decoding goes on, the instruction fetch unit fetches the next

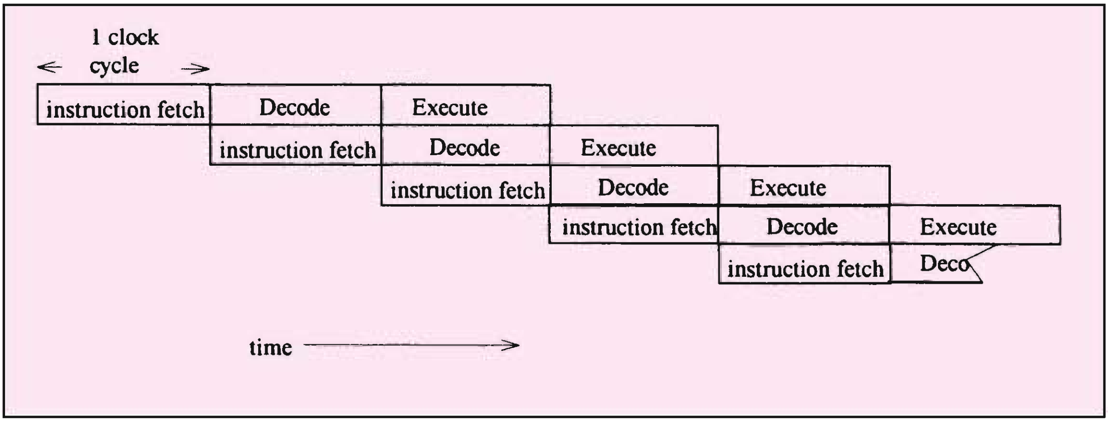

Figure 1 A 3-stage instruction pipeline.

instruction. When the current instruction is being executed by the execution unit, the decoding unit decodes the next instruction and so on. Note that this means that the processor executes one instruction every clock cycle. Hence, ideally all units are busy all the time and this process is referred to as pipelining. This is shown in *Figure 1*, where a simple pipeline of three units is illustrated.

We draw the following analogy: In a bread-factory assembly line, one person fetches the bread, the one next to him slices the bread with a slicing machine, the next one packs it, and the last one stores the packed bread in the shelf and maintains the count. Of course, both in this situation and in the processor, all units of the pipeline must take the same amount of time (typically one clock cycle) to do their bit. If this is not done, the units taking less time will remain partly idle in every clock cycle. This may be desirable (and humane!) in a human pipeline but will lead to performance degradation due to under-utilization of the processor. Hence, designers are forced to keep the decoding unit (which is usually the most time consuming process) simple. Keeping the decoding unit simple means having a reduced number of instructions. This is the idea behind the RISC (reduced instruction set computer) as opposed to CISC (complex instruction set computer). The 8085 and 8086 for instance, are CISC implementations.

The most important input for a processor is the instructions it fetches. The main steps involved in this process are instruction fetch, decode and execute.

We have spoken about three basic units: *fetch, decode* and *execute.*

The P6 contains fourteen such independent units (called stages), as compared to the Pentium's five stages. Hence it is said to be *super-pipelined.* In addition to pipelining, the independence of the units such as the integer units (two of them), floating-point unit and address generator units allows the P6 to simultaneously execute up to five instructions per clock cycle instead of one in the case of the 80486 and two in the case of the Pentium. This is why the P6 is called *super-scalar.* To further help this feature, the P6 supports out-of-order execution of the sequential instruction stream (i.e., re-orders them) such that the independent units are all kept busy. This is done without affecting the integrity of the program. In other words, the designers have done their best to build-in circuitry to try and keep all the units busy all the time. This does not always happen because programs often contain instructions that depend on the previous instruction. Instruction B for instance is said to be dependent on instruction A, if the execution of B depends on the result of the execution of A.

The P6 contains 14 independent units (called stages), as compared to the Pentium's five stages.

Branch Prediction

We have seen the pipe-lined structure of the P6. We also know that the processor fetches its instructions one-by-one from successive memory locations. This is known as the program flow. Often, a break occurs in the program flow. This means the processor must begin fetching instructions from some other location (rather than a successive location) and continue fetching from that location. These breaks are referred to as 'jumps' since the processor jumps to a new location to continue its operation. Jumps occur whenever there is a conditional instruction or when there are loops in a program. This is illustrated below:

The independence of the units allows the P6 to simultaneously execute up to five instructions per clock cycle instead of one in the case of the 80486 and two in the case of Pentium.

```
1.  if number A is greater than number B
2.     then multiply A by number C
3.  else multiply number B by number C
4.  fetch number D.
```

Features of the P6 Processor

- Has a super-pipeline consisting of 14 stages.
- First version operates on a 133 Mhz clock and many instructions take a single clock cycle to execute.
- Super-scalar architecture capable of issuing upto 3 instructions at a time.
- 5 parallel execution units: 2 integer, one load, one store and one floating point unit (FPU).
- Supports out-of-order execution of instructions.
- One internal cache of 16 Kbytes (8 kbytes for code and 8 kbytes for data).
- A second cache (on a separate die but on the same package) of 256 kbytes with a separate 64-bit data path.
- P6 is about 800 times faster than a 8086 and about 2.5 times faster than a Pentium.
- Dissipates 20 watts of power.
- Estimated initial price : $1500.

Now, while executing step 1 if the processor finds that number A is indeed greater than number B it will fetch successive instructions of step 2 (no jump occurs). But after executing step 2 it will jump to step 4 and there is a break. If the processor finds at step 1 that number A is less than number B then it jumps to step 3, but no jump occurs between step 3 and step 4.

An example of a loop where the program calculates the power of 2 given the exponent 'n' is as follows:

```
1.  count=0, answer=1
2.  do steps 3 and 4 until count = n
3.    answer= 2*answer
4.    increment count
5.  store answer
```

Until the count is 'n' the program will jump back from step 4 to step 2 (where the value of count is checked).

Every time there is a break in program flow the overhead on the instruction fetch unit to go to a new location and start fetching again is quite high and this leads to 'bubbles' in the pipeline. The

Bubbles are formed in the pipeline whenever there is a break in program flow.

bubbles refer to the fact that some units will be idle for one or more clock cycles depending on the overhead. Let us go back to our bread-factory analogy. Notice that the person who fetches the bread would rather go and fetch many loaves of bread (from the oven) than only one. That is because of the overhead he incurs to go, pick and return — he might as well pick up several rather than one. This is exactly what happens in a processor. It is far more efficient to fetch more than one instruction at a time from memory and store them in an 'instruction queue' in readiness for the decoding unit. This is called 'pre-fetching'. Let us suppose that the factory produces two varieties of bread (milk bread and sweet bread) and the last person in the pipeline (who is also keeping count) tells the first one: " Enough of milk breads, we need to produce sweet breads now". The first person has go to another oven and fetch the sweet breads. Till that time the others will be idle. On the other hand if the last person can predict in advance and accordingly inform the first person to fetch sweet breads, then there won't be bubbles and no person will be idle. The branch prediction unit in the P6 does something similar. It uses statistical techniques to predict whether a (jump) branch will be taken or not and accordingly does the prefetching to avoid bubbles. Since this is probabilistic, there are times when the predictions are not right and there is a temporary dip in performance.

The branch prediction unit in the P6 uses statistical techniques to predict whether a (jump) branch will be taken or not and accordingly does the pre-fetching to avoid bubbles.

Caching

The powerful computing engine capable of high speed data processing has to be supplied with data at the required rate. This is difficult to do if one considers the memory technology and the delays introduced by external electrical connections. To achieve this, they have built two caches for the chip; (caching was explained separately in an earlier article of the series). One of these (the level-one or L1 cache) is 16 kilobytes in size, and is integrated with the processor core on the same silicon die which consists of around 5.5 million transistors. The second cache (L2 cache) is 256 kilobytes in size and built on a separate silicon die consisting of 15.5 million transistors. The two dies are then

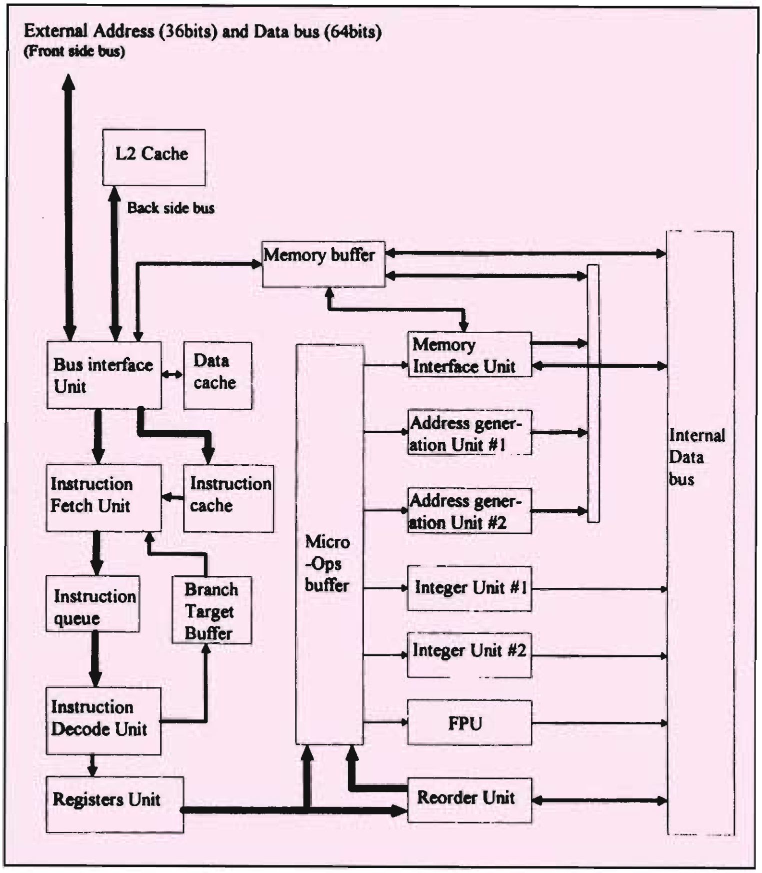

***Figure 2** The internal structure of the P6 processor.*

The P6 can run 16-bit software but will do so in an inefficient manner. In fact the performance of P6 will be slower than the Pentium for 16bit software!

integrated on the same ceramic package and bonded together by a technique called wire-bonding. This package, which has a total of 387 pins, is known as an MCM (multichip module). The L2 cache has its own path to the bus interface unit of the processor, separate from the path to the main memory. Both these data paths are 64 bits wide. This sophisticated design of caches ensures that the data is supplied to the P6 at a rate that matches the consumption rate of the processor. The existence of the L2 cache also saves system designers (designing boards with the P6 as the CPU) the trouble of building an external cache system and thus simplifies board design considerably. There is one caveat however to all this performance enhancement. The P6 designers expected that by the time it was released, users would be predominantly running 32-bit software and so they optimized the architecture for this kind

Intel has already started development of the P7, which will push performance levels even further.

of software. The P6 can run 16-bit software but will do so in an inefficient manner. In fact, the performance of P6 will be slower than the Pentium for 16-bit software! This is the price that Intel has to pay to carry the 'DOS baggage' (see an earlier article in ***Resonance*** Vol.1, No.1, on Windows 95) in all its processors.

Conclusion

Although P6 is a high-performance microprocessor with a feature rich architecture, it has competition from other processors such as AMD's K5, NexGen's Nx586 and Cyrix's 5x86 which run x86 code and all of which share architectural innovations similar to the P6. The markets will ultimately decide which one succeeds. Meanwhile, Intel has started development of the P7, which will boost performance levels even further.

Address for correspondence
Vijnan Shastri
Centre for Electronics
Design Technology,
Indian Institute of Science,
Bangalore 560 012, India.

Suggested Reading

'P6 the Next Step?'. *PC Magazine*. September 12, 1995.
'Intel's P6'. *Byte* . April 1995.

NOW DOES THIS EXPLAIN THE CHOICE OF THE WALL PAPER, SIR ?

MOHAN DEVADAS