Advanced CISC Implementations: Pentium 4

Krste Asanovic
Laboratory for Computer Science
M.I.T.

http://www.csg.lcs.mit.edu/6.823

Intel Pentium Pro (1995)

- During decode, translate complex x86 instructions into RISC-like micro-operations (uops)
  – e.g., “R ← R op Mem” translates into

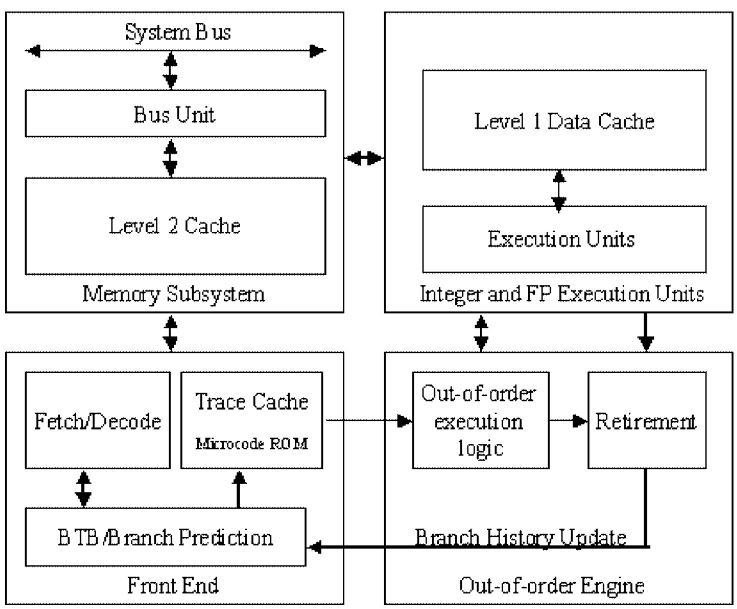
    load T, Mem   # Load from Mem into temp reg
    R ← R op T   # Operate using value in temp
- Execute uops using speculative out-of-order superscalar engine with register renaming
- Pentium Pro family architecture (P6 family) used on Pentium-II and Pentium-III processors

Intel Pentium 4 (2000)

- **Deeper pipelines than P6 family**
  - about half as many levels of logic per pipeline stage as P6
- **Trace cache holds decoded uops**
  - only has a single x86->uop decoder
- **Decreased latency in same process technology**
  - aggressive circuit design
  - new microarchitectural tricks

This lecture contains figures and data taken from: “The microarchitecture of the Pentium 4 processor”, Intel Technology Journal, Q1, 2001

Pentium 4 Block Diagram



P-III vs. P-4 Pipelines

Basic Pentium® III Processor Misprediction Pipeline

1	2	3	4	5	6	7	8	9	10
Fetch	Fetch	Decode	Decode	Decode	Rename	ROB Rd	Rdy/Sch	Dispatch	Exec

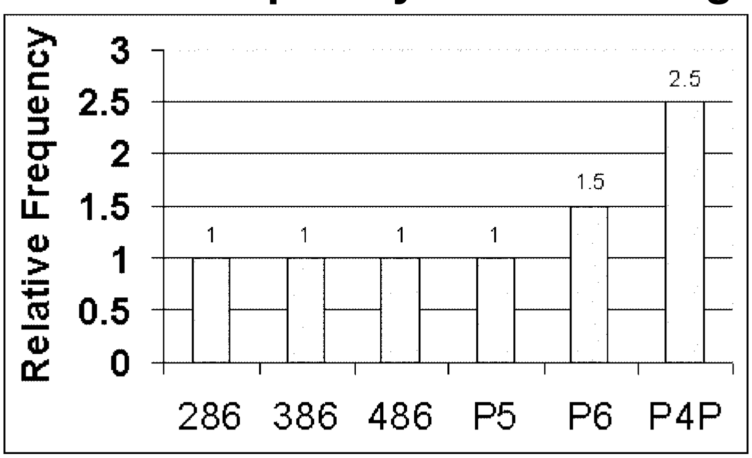
Basic Pentium® 4 Processor Misprediction Pipeline

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
TC Nxt IP		TC Fetch		Drive	Alloc	Rename		Que	Sch	Sch	Sch	Disp	Disp	RF	RF	Ex	Flgs	Br Ck	Drive

- **In same process technology, ~1.5x clock frequency**
- **Performance Equation:**

$$\frac{\text{Time}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} * \frac{\text{Cycles}}{\text{Instruction}} * \frac{\text{Time}}{\text{Cycle}}$$

Relative Frequency of Intel Designs



- **Over time, fewer logic levels per pipeline stage and more advanced circuit design**
- **Higher frequency in same process technology**

Deep Pipeline Design

Greater potential throughput but:

- **Clock uncertainty and latch delays eat into cycle time budget**
  - doubling pipeline depth gives less than twice frequency improvement
- **Clock load and power increases**
  - more latches running at higher frequencies
- **More complicated microarchitecture needed to cover long branch mispredict penalties and cache miss penalties**
  - from Little's Law, need more instructions in flight to cover longer latencies ➔ larger reorder buffers
- **P-4 has three major clock domains**
  - Double pumped ALU (3 GHz), small critical area at highest speed
  - Main CPU pipeline (1.5 GHz)
  - Trace cache (0.75 GHz), save power

Pentium 4 Trace Cache

- **Holds decoded uops in predicted program flow order, 6 uops per line**

Code in memory

```
    cmp
    br T1
    ...
T1: sub
    br T2
    ...
T2: mov
    sub
    br T3
    ...
T3: add
    sub
    mov
    br T4
    ...
T4:
```

→

Code packed in trace cache (6 uops/line)

cmp	br T1	sub
br T2	mov	sub

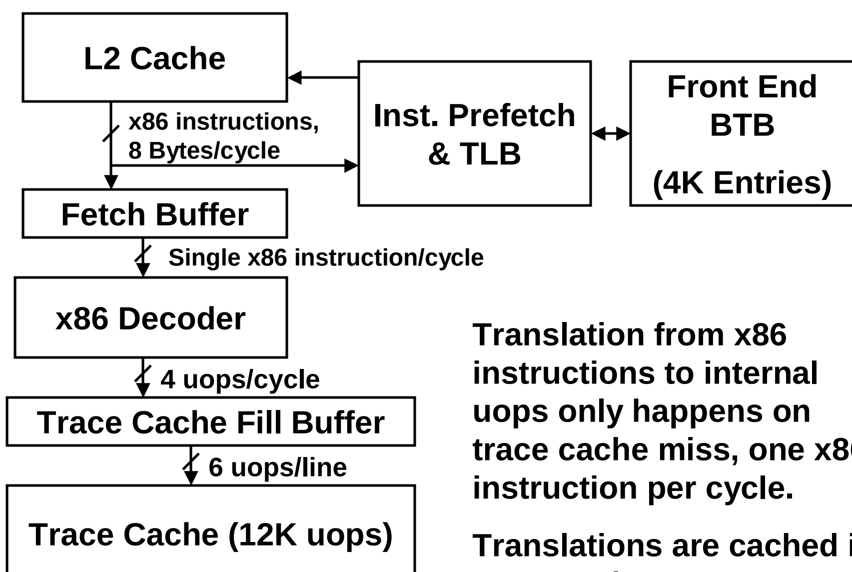
br T3	add	sub
mov	br T4	T4:...

Trace cache fetches one 6 uop line every 2 CPU clock cycles (runs at ½ main CPU rate)

Trace Cache Advantages

- **Removes x86 decode from branch mispredict penalty**
  - Parallel x86 decoder took 2.5 cycles in P6, would be 5 cycles in P-4 design
- **Allows higher fetch bandwidth fetch for correctly predicted taken branches**
  - P6 had one cycle bubble for correctly predicted taken branches
  - P-4 can fetch a branch and its target in same cycle
- **Saves energy**
  - x86 decoder only powered up on trace cache refill

Pentium 4 Front End

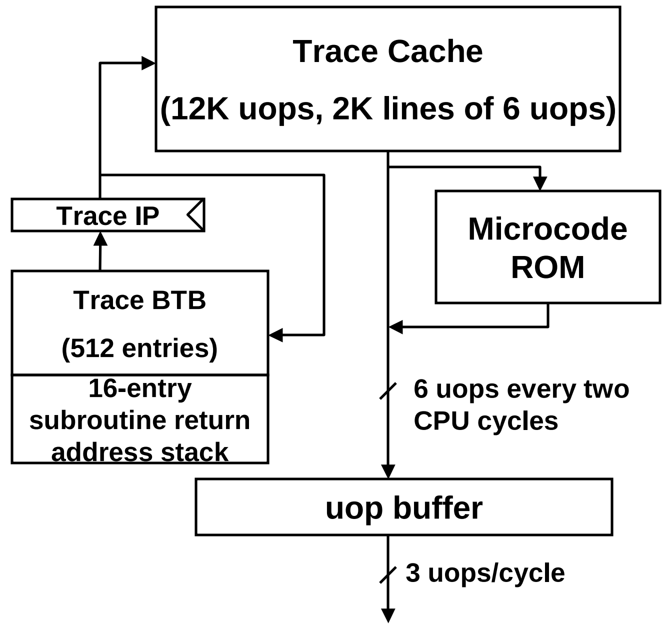


Translation from x86 instructions to internal uops only happens on trace cache miss, one x86 instruction per cycle.

Translations are cached in trace cache.

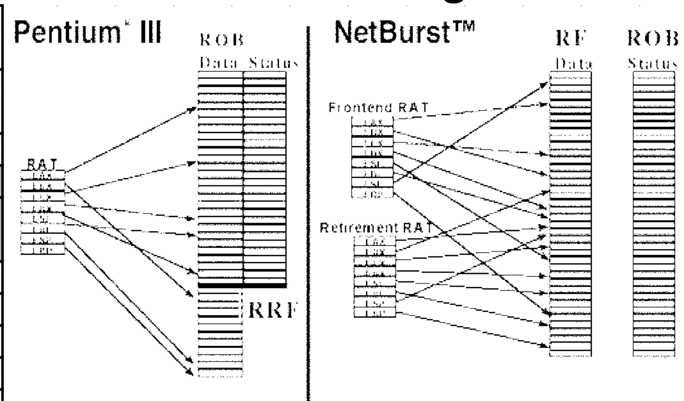
P-4 Trace Cache Fetch

Stage	
1	TC Next IP (BTB)
2	
3	TC Fetch
4	
5	Drive
6	Alloc
7	Rename
8	
9	Queue
10	Schedule 1
11	Schedule 2
12	Schedule 3
13	Dispatch 1
14	Dispatch 2
15	Register File 1
16	Register File 2
17	Execute
18	Flags
19	Branch Check
20	Drive



P-III vs. P-4 Renaming

Stage	
1	TC Next IP (BTB)
2	
3	TC Fetch
4	
5	Drive
6	Alloc
7	Rename
8	
9	Queue
10	Schedule 1
11	Schedule 2
12	Schedule 3
13	Dispatch 1
14	Dispatch 2
15	Register File 1
16	Register File 2
17	Execute
18	Flags
19	Branch Check
20	Drive



P-4 physical register file separated from ROB status.

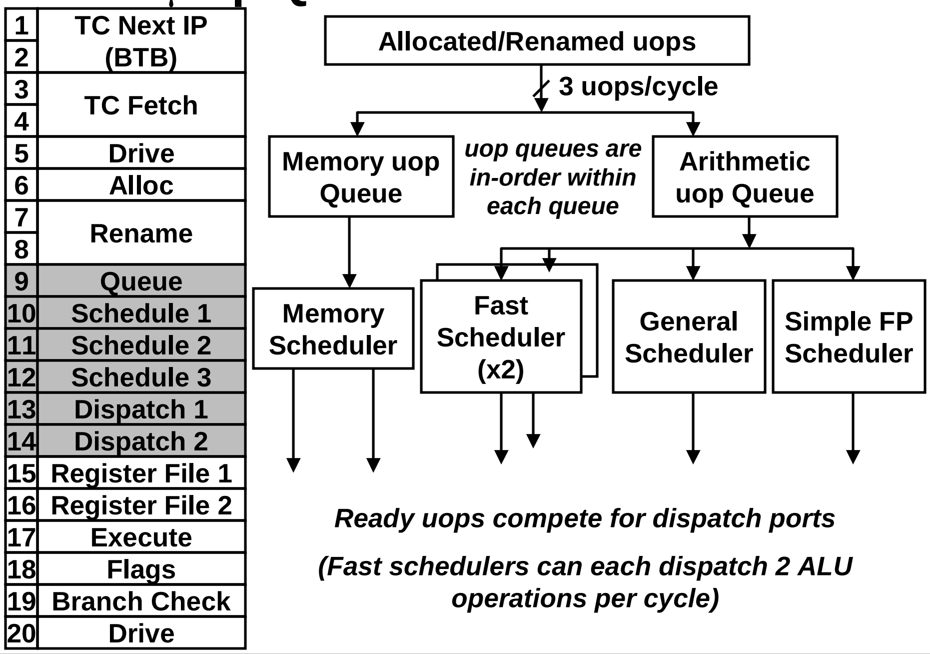
ROB entries allocated sequentially as in P6 family.

One of 128 physical registers allocated from free list.

No data movement on retire, only Retirement RAT updated.

Krste Asanovic

P-4 μOp Queues and Schedulers

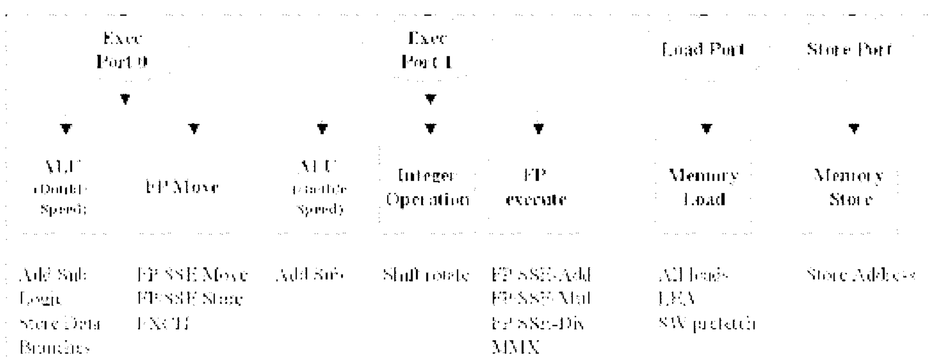


Ready uops compete for dispatch ports

(Fast schedulers can each dispatch 2 ALU operations per cycle)

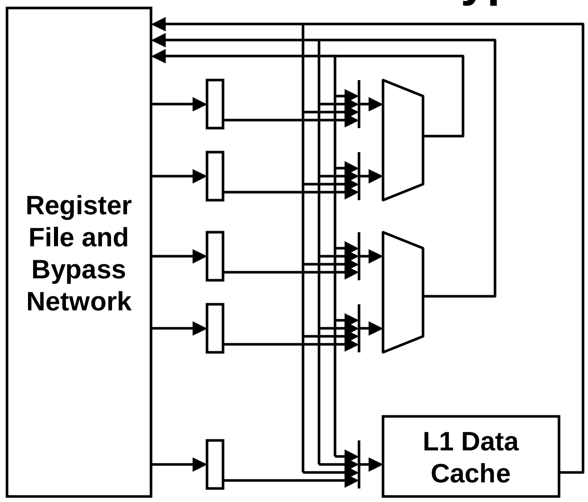
Krste Asanovic

P-4 Execution Ports



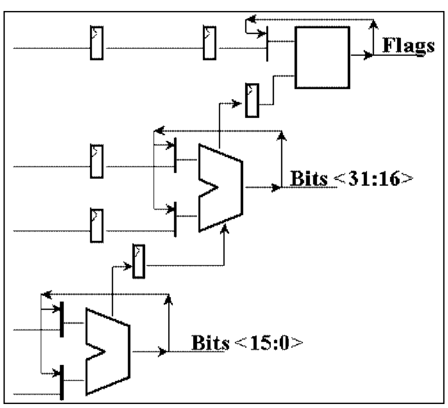
- **Schedulers compete for access to execution ports**
- **Loads and stores have dedicated ports**
- **ALUs can execute two operations per cycle**
- **Peak bandwidth of 6 uops per cycle**
  - **load, store, plus four double-pumped ALU operations**

P-4 Fast ALUs and Bypass Path



- Fast ALUs and bypass network runs at double speed
- All "non-essential" circuit paths handled out of loop to reduce circuit loading (shifts, mults/divs, branches, flag/ops)
- Other bypassing takes multiple clock cycles

P-4 Staggered ALU Design



- Staggers 32-bit add and flag compare into three ½ cycle phases
  - low 16 bits
  - high 16 bits
  - flag checks
- Bypass 16 bits around every ½ cycle
  - back-back dependent 32-bit adds at 3GHz in 0.18μm
- L1 Data Cache access starts with bottom 16 bits as index, top 16 bits used as tag check later

P-4 Load Schedule Speculation

1	TC Next IP
2	
3	TC Fetch
4	
5	Drive
6	Alloc
7	Rename
8	
9	Queue
10	Schedule 1
11	Schedule 2
12	Schedule 3
13	Dispatch 1
14	Dispatch 2
15	Register File 1
16	Register File 2
17	Load Execute 1
18	Load Execute 2
19	Branch Check
20	Drive

Long delay from schedulers to load hit/miss

- P-4 guesses that load will hit in L1 and schedules dependent operations to use value
- If load misses, only dependent operations are replayed

P-4 Branch Penalty

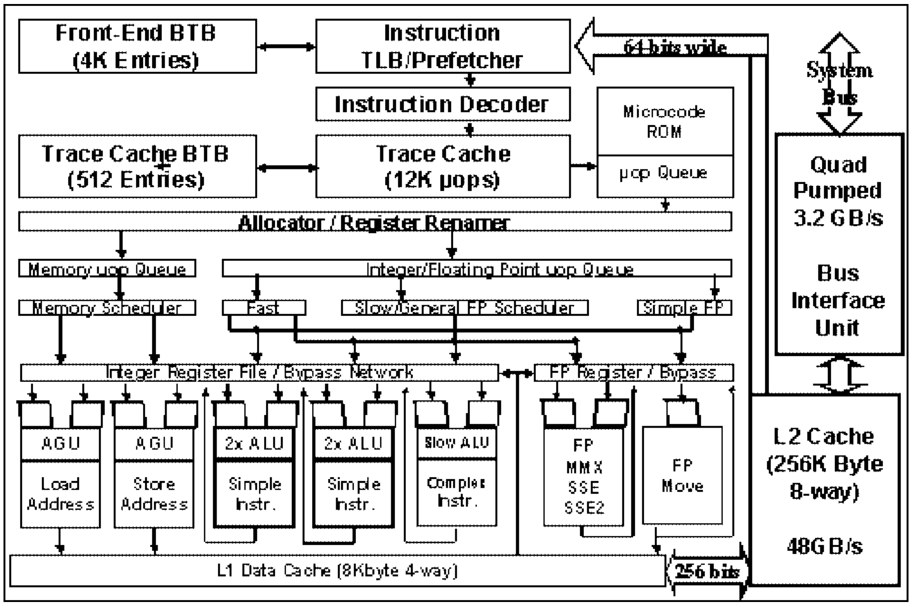
1	TC Next IP
2	
3	TC Fetch
4	
5	Drive
6	Alloc
7	Rename
8	
9	Queue
10	Schedule 1
11	Schedule 2
12	Schedule 3
13	Dispatch 1
14	Dispatch 2
15	Register File 1
16	Register File 2
17	Execute
18	Flags
19	Branch Check
20	Drive

20 cycle branch mispredict penalty

- P-4 uses new “trade secret” branch prediction algorithm
- Intel claims 1/3 fewer mispredicts than P6 algorithm

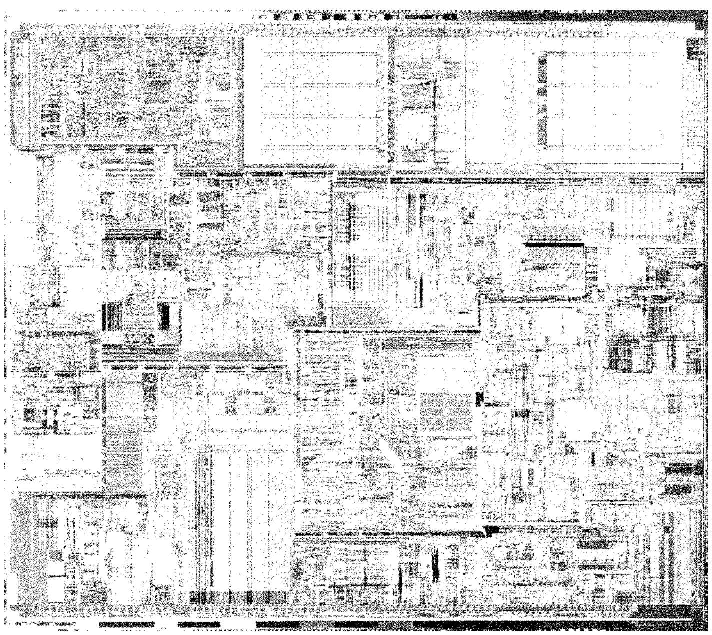
P-4 Microarchitecture

Krste Asanovic
May 7, 2001
6.823, L22--19



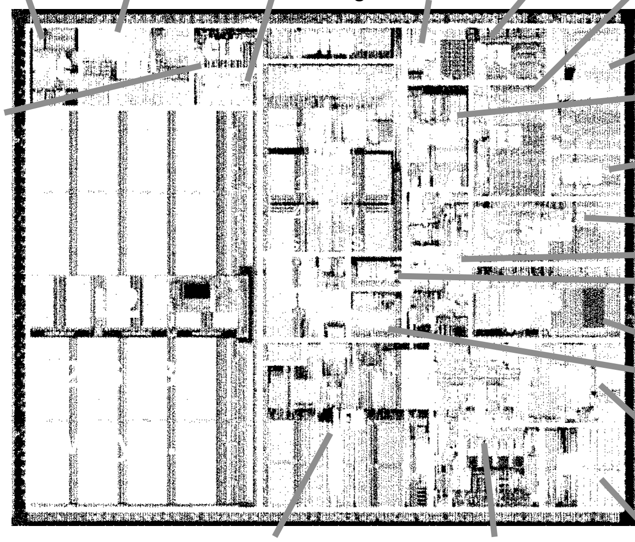
Pentium-4 Die Photo

Krste Asanovic
May 7, 2001
6.823, L22--20



Krste Asanovic
May 7, 2001
6.823, L22--21

Pentium-III Die Photo



Krste Asanovic
May 7, 2001
6.823, L22--22

Scaling of Wire Delay

- Over time, transistors are getting relatively faster than long wires
  - wire resistance growing dramatically with shrinking width and height
  - capacitance roughly fixed for constant length wire
  - RC delays of fixed length wire rising
- Chips are getting bigger
  - P-4 >2x size of P-III
- Clock frequency rising faster than transistor speed
  - deeper pipelines, fewer logic gates per cycle
  - more advanced circuit designs (each gate goes faster)

$\Rightarrow$ Takes multiple cycles for signal to cross chip

Visible Wire Delay in P-4 Design

1	TC Next IP
2	
3	TC Fetch
4	
5	Drive
6	Alloc
7	Rename
8	
9	Queue
10	Schedule 1
11	Schedule 2
12	Schedule 3
13	Dispatch 1
14	Dispatch 2
15	Register File 1
16	Register File 2
17	Execute
18	Flags
19	Branch Check
20	Drive

Pipeline stages dedicated to just driving signals across chip!

Instruction Set Translation

- **Convert a target ISA into a host machine's ISA**
- **Pentium Pro (P6 family)**
  - translation in hardware after instruction fetch
- **Pentium-4 family**
  - translation in hardware at level 1 instruction cache refill
- **Transmeta Crusoe**
  - translation in software using "Code Morphing"