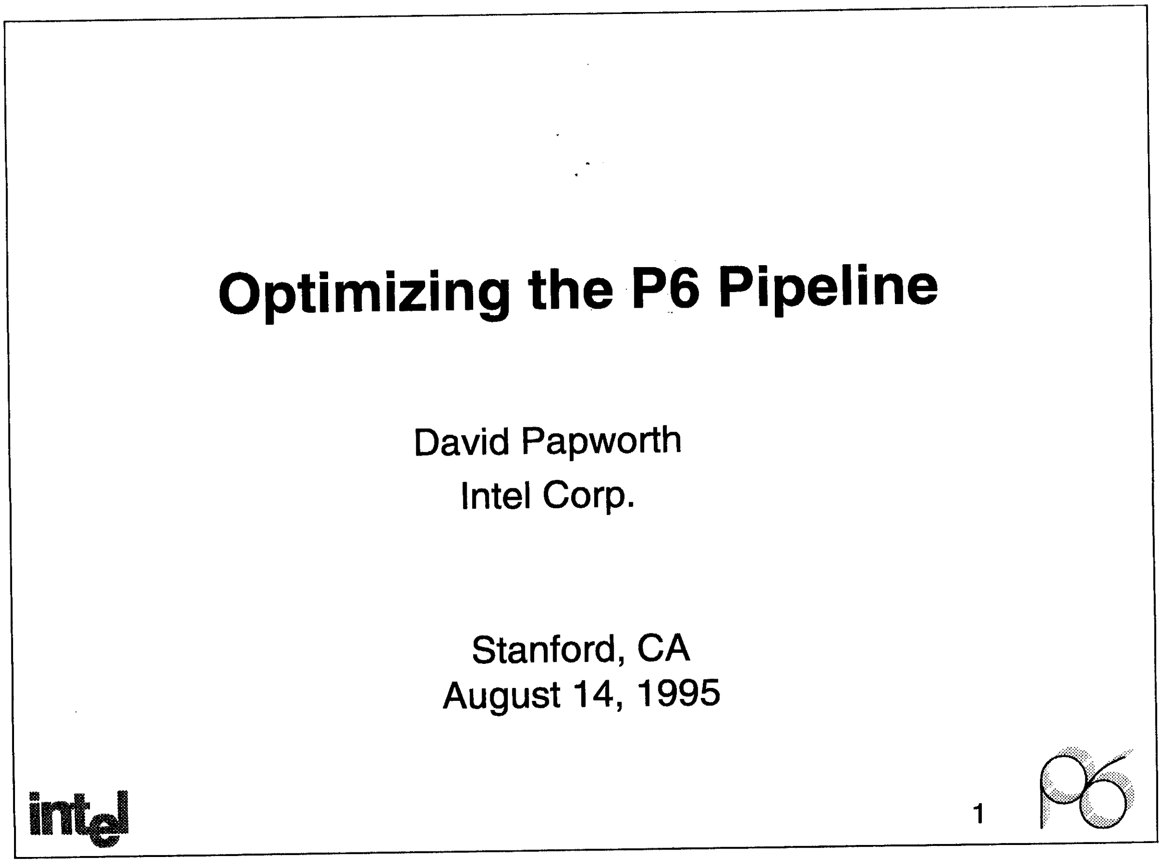
2.1-01



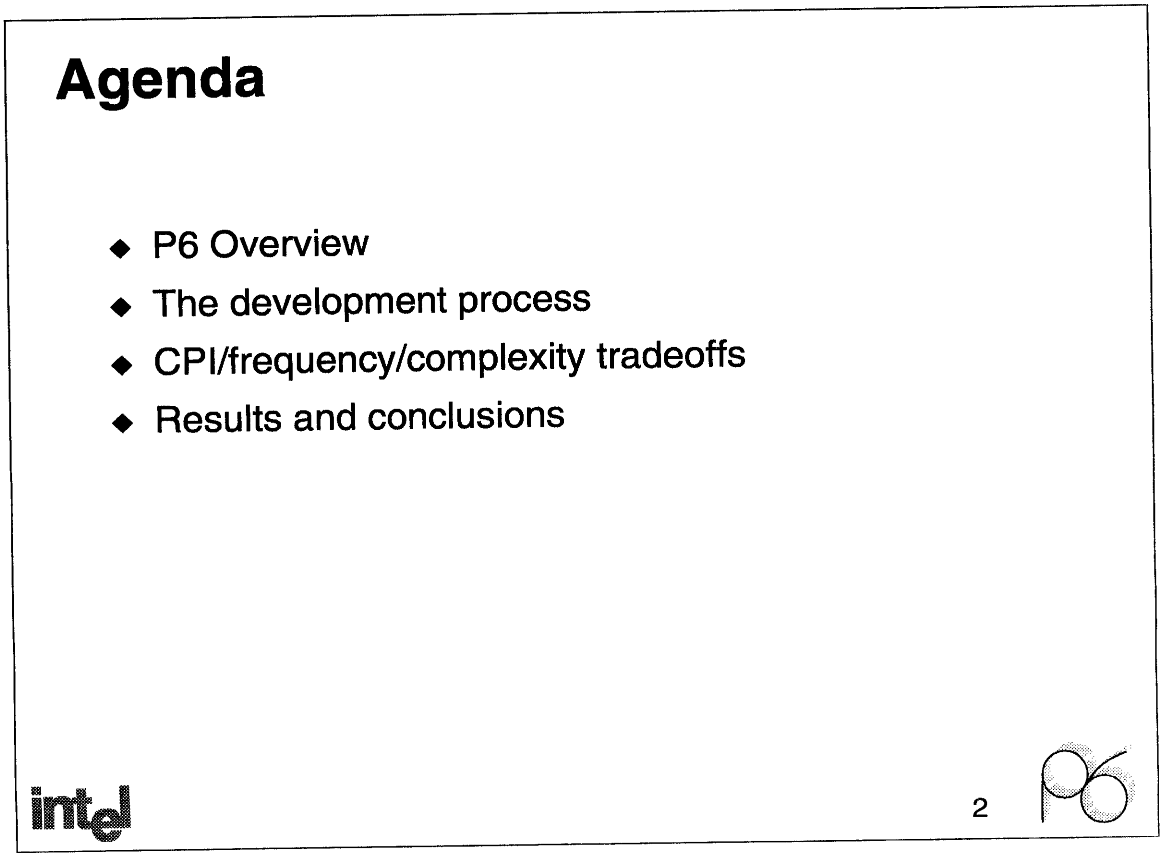
Optimizing the P6 Pipeline

David Papworth
Intel Corp.

Stanford, CA
August 14, 1995

1

2.1-02



Agenda

- P6 Overview
- The development process
- CPI/frequency/complexity tradeoffs
- Results and conclusions

2

2.1-03

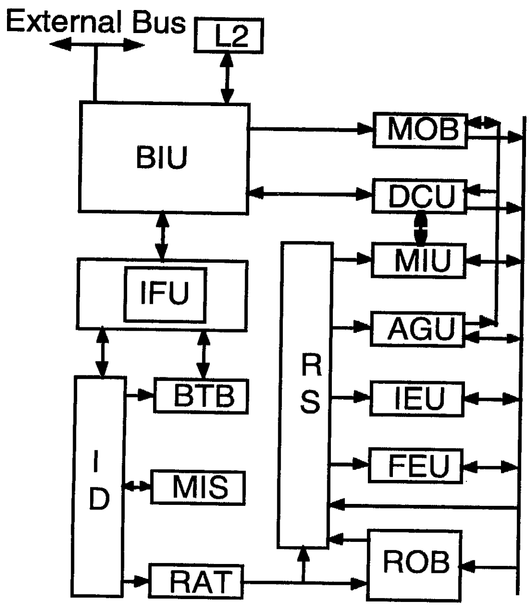
Technology profile

- 32-bit Intel Architecture processor
- Dynamic Execution microarchitecture
  - Speculative and Out-of-order Execution
  - Micro-dataflow
  - Superscalar
  - Superpipelined
- 8K/8KB non-blocking L1 caches
- 256KB integrated non-blocking L2 cache

intel 3 P6

2.1-04

Implementation: Microarchitecture

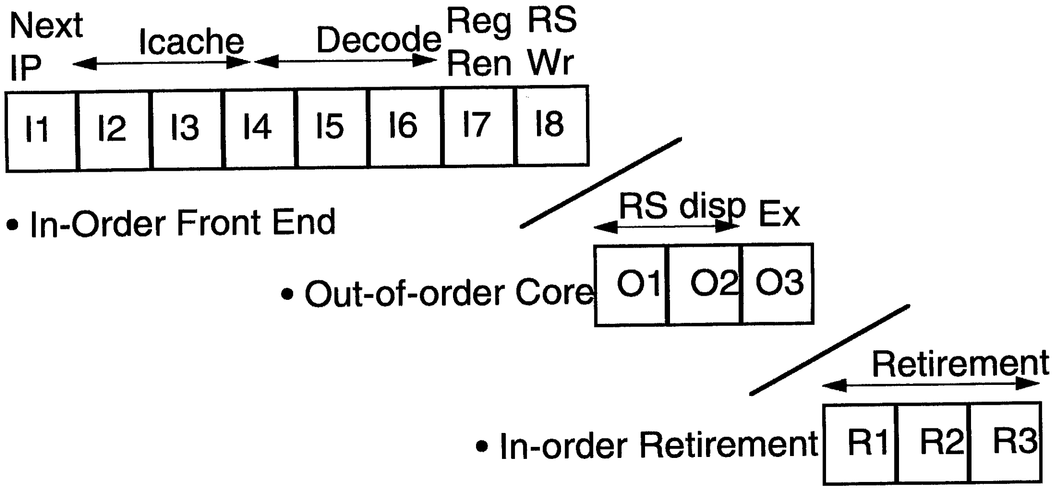


- In-Order Front End
- Out-of-order Core
- In-Order Retire

intel 4 P6

2.1-05

Implementation: Pipeline



Next IP ◄──Icache──►◄──Decode──► Reg Ren RS Wr

I1	I2	I3	I4	I5	I6	I7	I8

- In-Order Front End

◄──RS disp──► Ex

- Out-of-order Core

O1	O2	O3

◄──Retirement──►

- In-order Retirement

R1	R2	R3

5

2.1-06

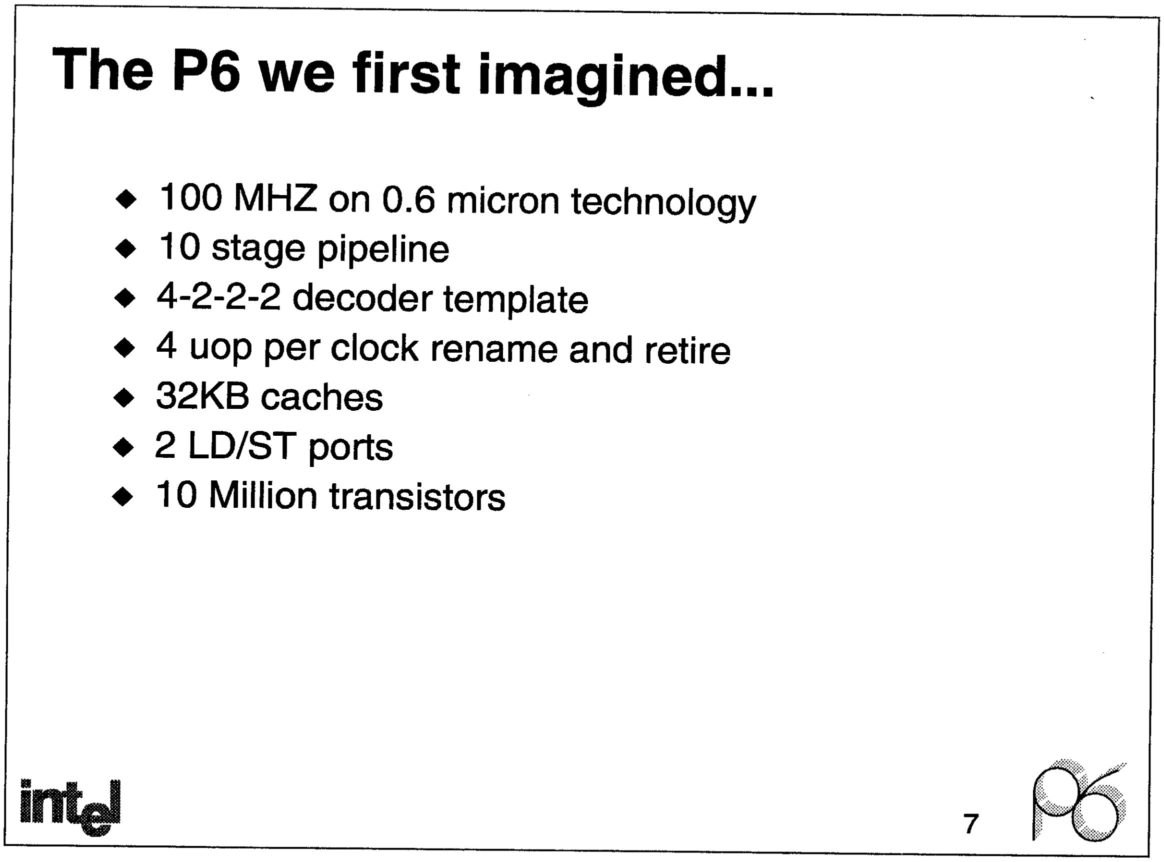
Typical goals for a major new CPU

(Relative to older design on same process)

- ◆ Achieve 1.5x to 2.0x the performance
- ◆ Run at about the same clock rate
- ◆ Use as many transistors as will fit
- ◆ Adopt new bus/cache/packaging to match higher core performance and add new capability (MP)

6

2.1-07

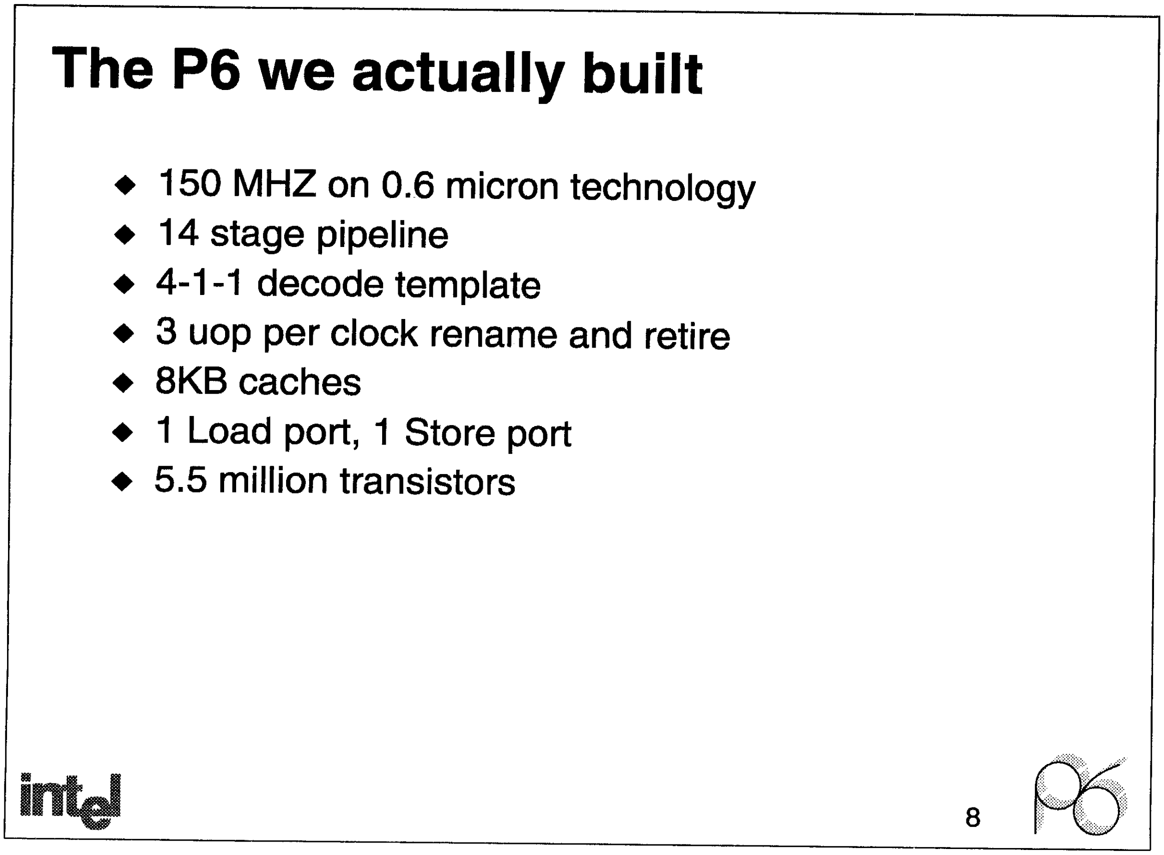


The P6 we first imagined...

- 100 MHZ on 0.6 micron technology
- 10 stage pipeline
- 4-2-2-2 decoder template
- 4 uop per clock rename and retire
- 32KB caches
- 2 LD/ST ports
- 10 Million transistors

7

2.1-08

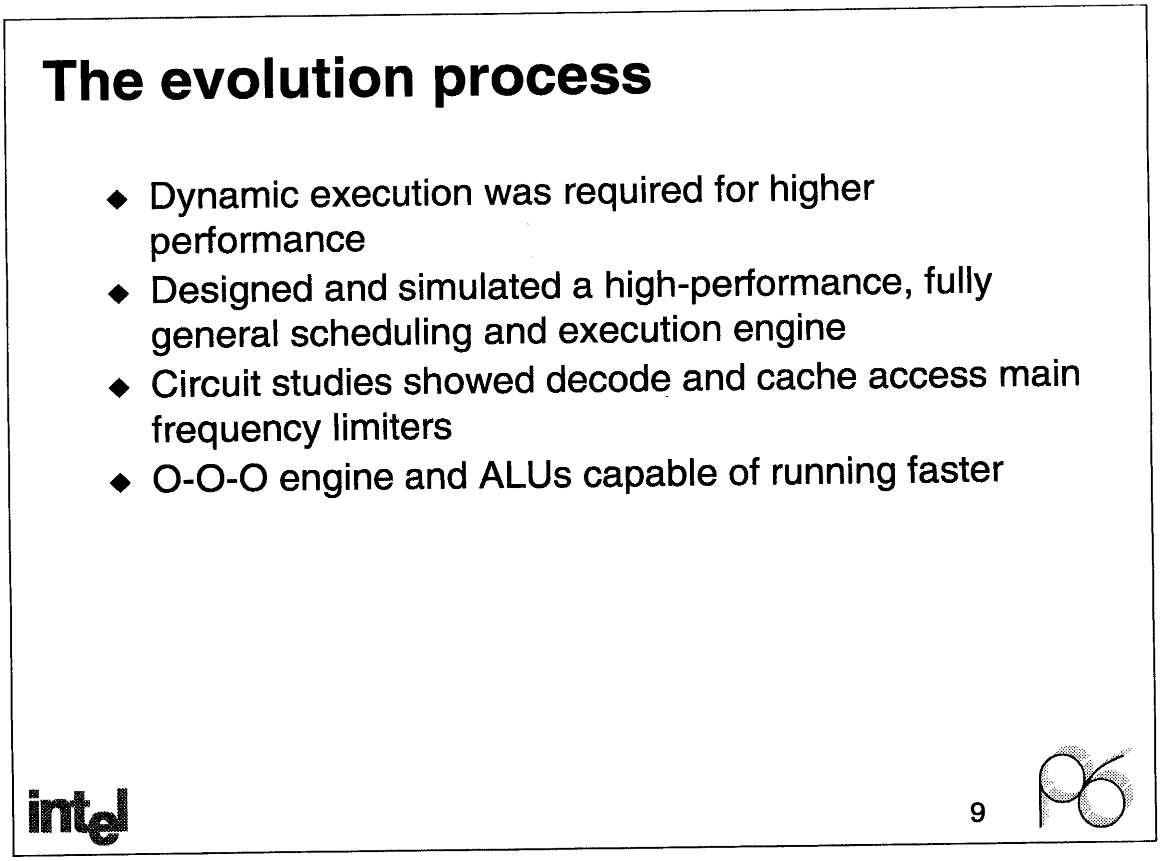


The P6 we actually built

- 150 MHZ on 0.6 micron technology
- 14 stage pipeline
- 4-1-1 decode template
- 3 uop per clock rename and retire
- 8KB caches
- 1 Load port, 1 Store port
- 5.5 million transistors

8

2.1-09

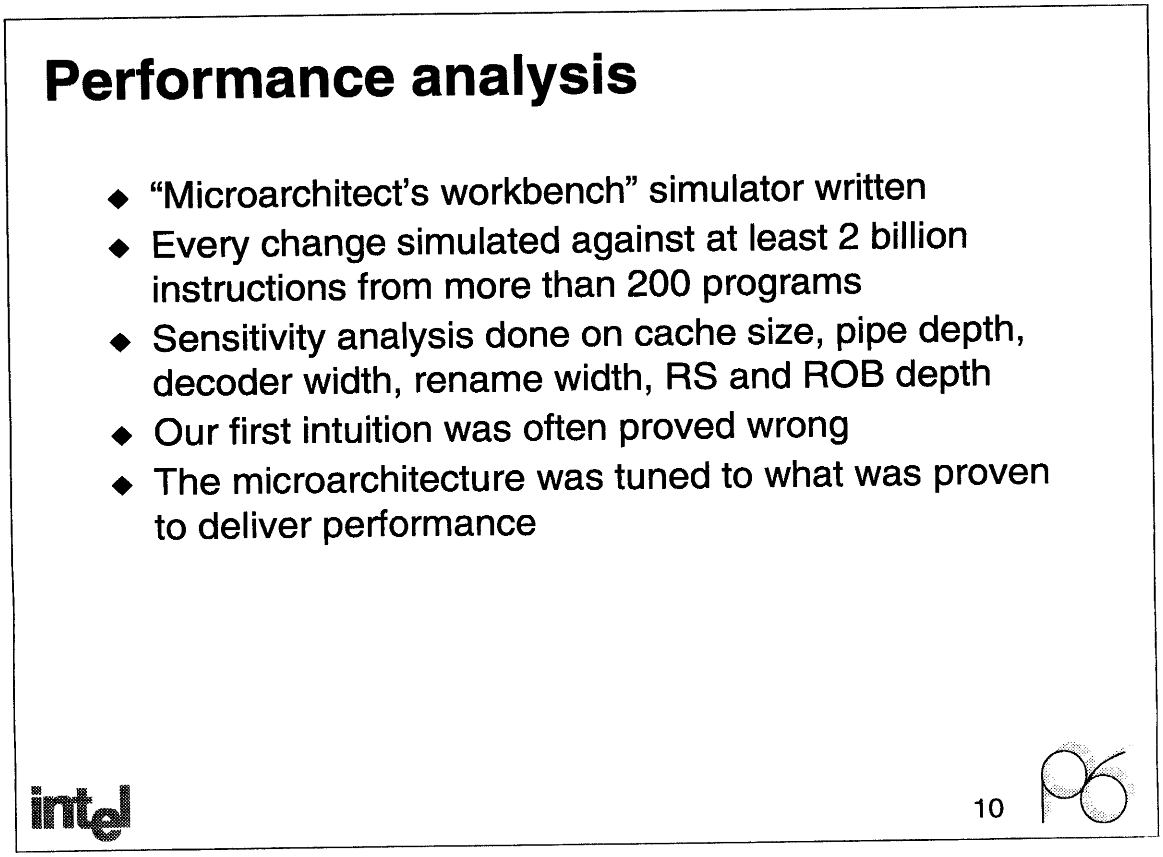


The evolution process

- Dynamic execution was required for higher performance
- Designed and simulated a high-performance, fully general scheduling and execution engine
- Circuit studies showed decode and cache access main frequency limiters
- O-O-O engine and ALUs capable of running faster

9

2.1-10



Performance analysis

- "Microarchitect's workbench" simulator written
- Every change simulated against at least 2 billion instructions from more than 200 programs
- Sensitivity analysis done on cache size, pipe depth, decoder width, rename width, RS and ROB depth
- Our first intuition was often proved wrong
- The microarchitecture was tuned to what was proven to deliver performance

10

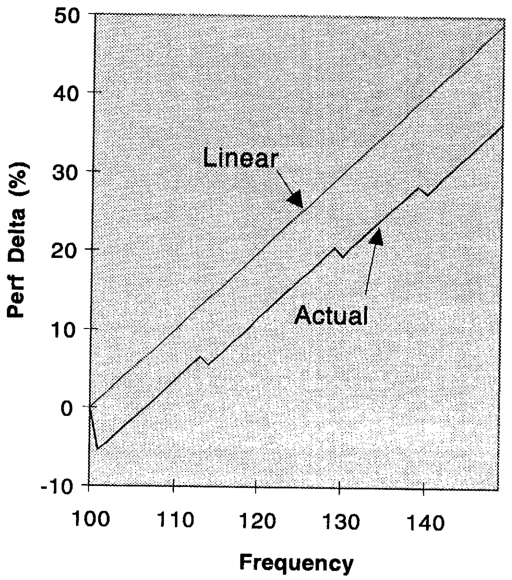
2.1-11

Optimizing CPI and Frequency

- Reduced gates per pipestage by 1/3
  - 50 % frequency gain
- Add 1 stage to data cache lookup
  - 7% CPI loss
- Add stages to front end
  - 3% CPI loss
- Add miscellaneous simplifications
  - 3% CPI loss
- Net result
  - 37% performance gain

11

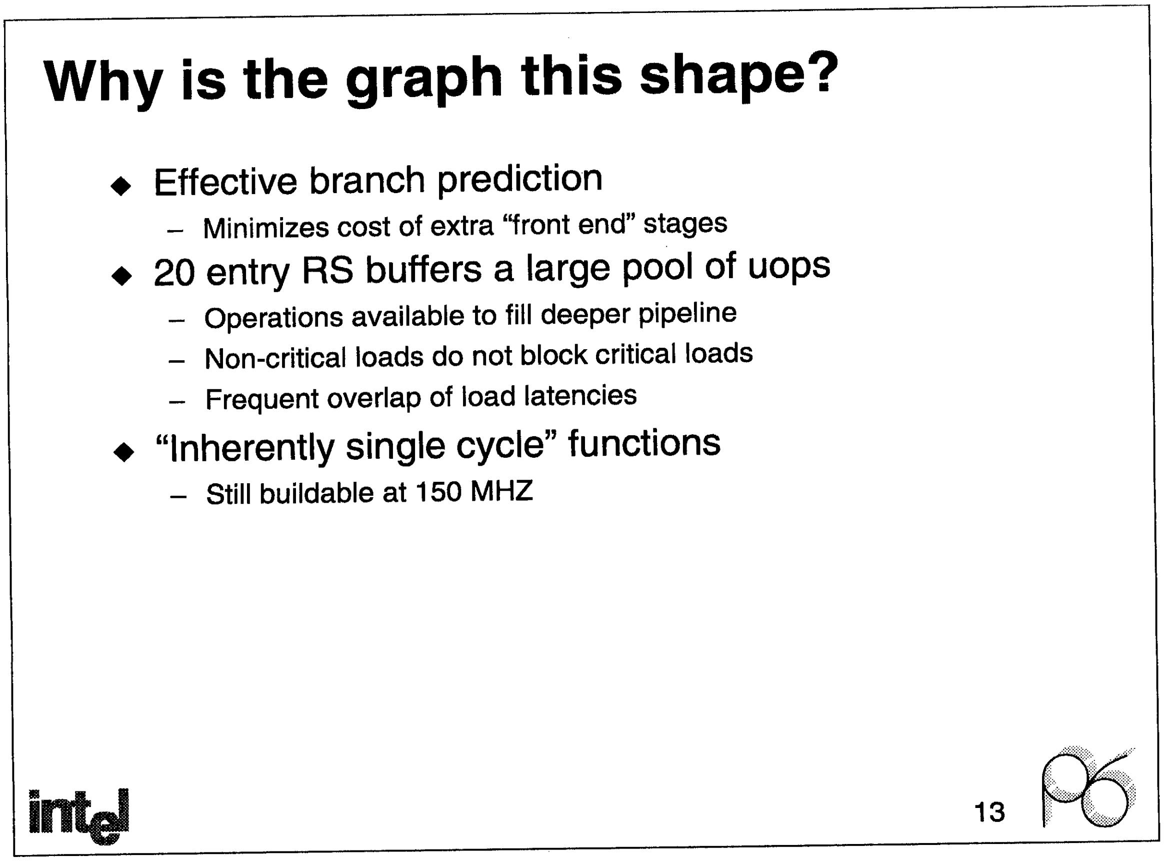
2.1-12



- Performance generally improves with frequency
- Must accept some CPI loss to enable higher frequency
- Net result is better performance (within limits)

12

2.1-13

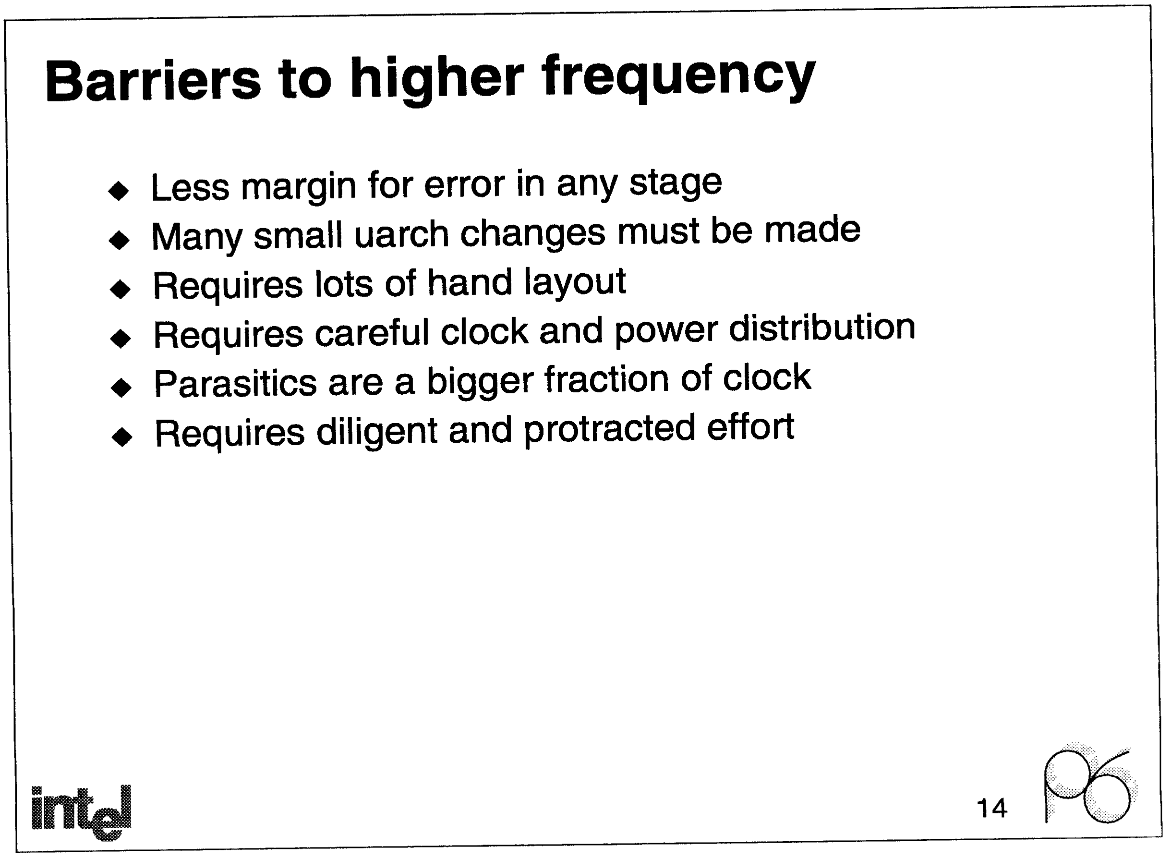


Why is the graph this shape?

- Effective branch prediction
  - Minimizes cost of extra “front end” stages
- 20 entry RS buffers a large pool of uops
  - Operations available to fill deeper pipeline
  - Non-critical loads do not block critical loads
  - Frequent overlap of load latencies
- “Inherently single cycle” functions
  - Still buildable at 150 MHZ

13

2.1-14



Barriers to higher frequency

- Less margin for error in any stage
- Many small uarch changes must be made
- Requires lots of hand layout
- Requires careful clock and power distribution
- Parasitics are a bigger fraction of clock
- Requires diligent and protracted effort

14

2.1-15

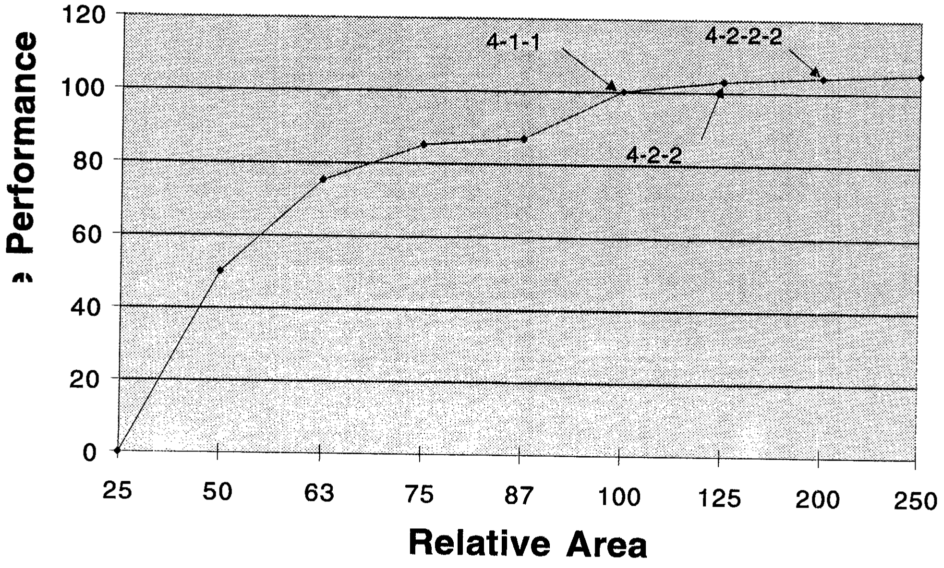
Optimizing complexity and area

- Most apps do not "ring the bell"
- Deep buffers and flexible scheduling allow time averaging of demand
- Remove wasteful overcapacity
  - Improves clock rate and die area
  - Reduces complexity and improves correctness
- Trim until slight performance loss

15

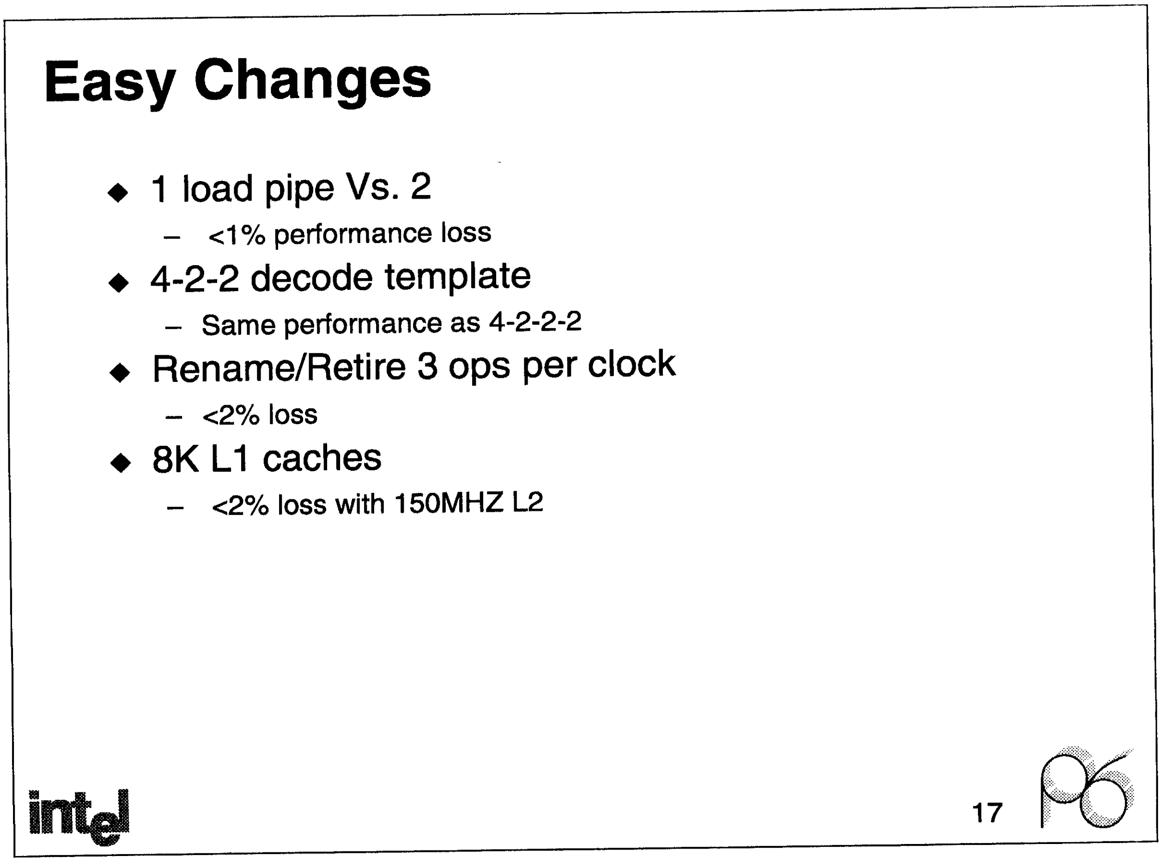
2.1-16

Area Vs. Performance for Decoder



16

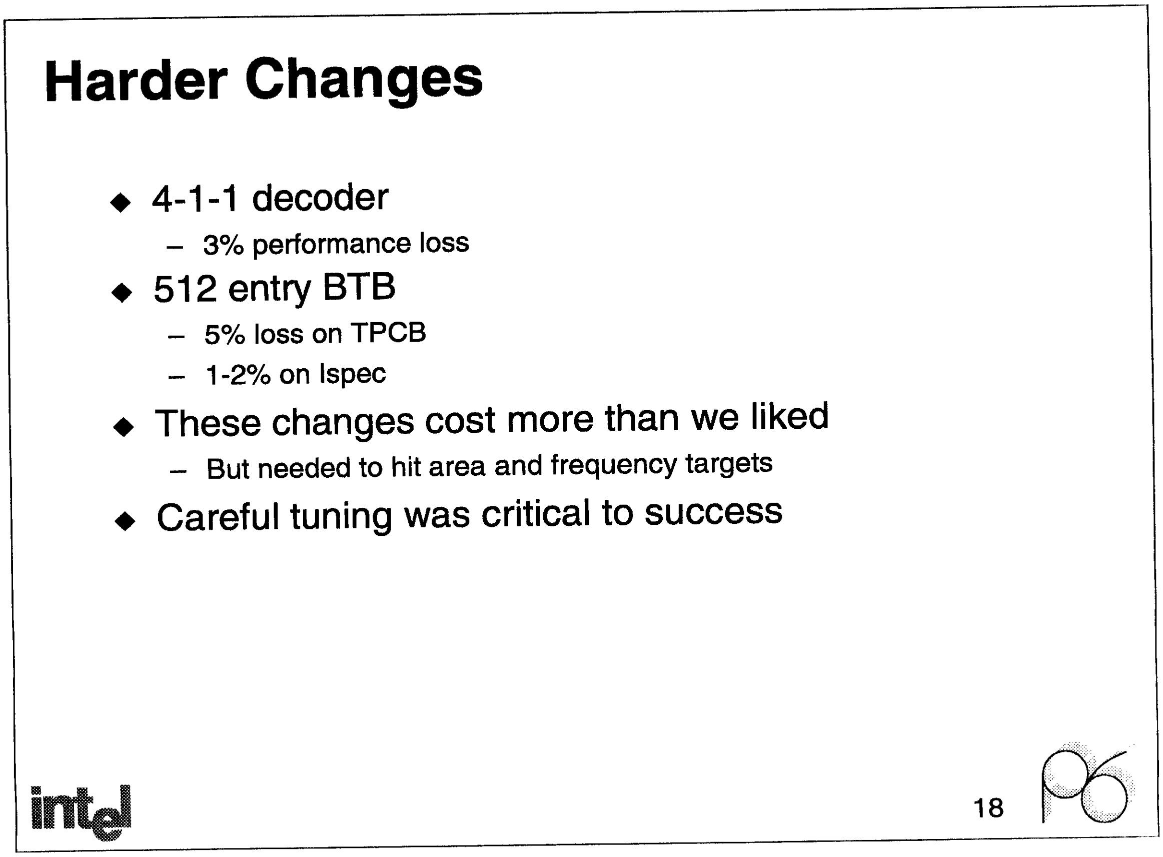
2.1-17



Easy Changes

- 1 load pipe Vs. 2
  - <1% performance loss
- 4-2-2 decode template
  - Same performance as 4-2-2-2
- Rename/Retire 3 ops per clock
  - <2% loss
- 8K L1 caches
  - <2% loss with 150MHZ L2

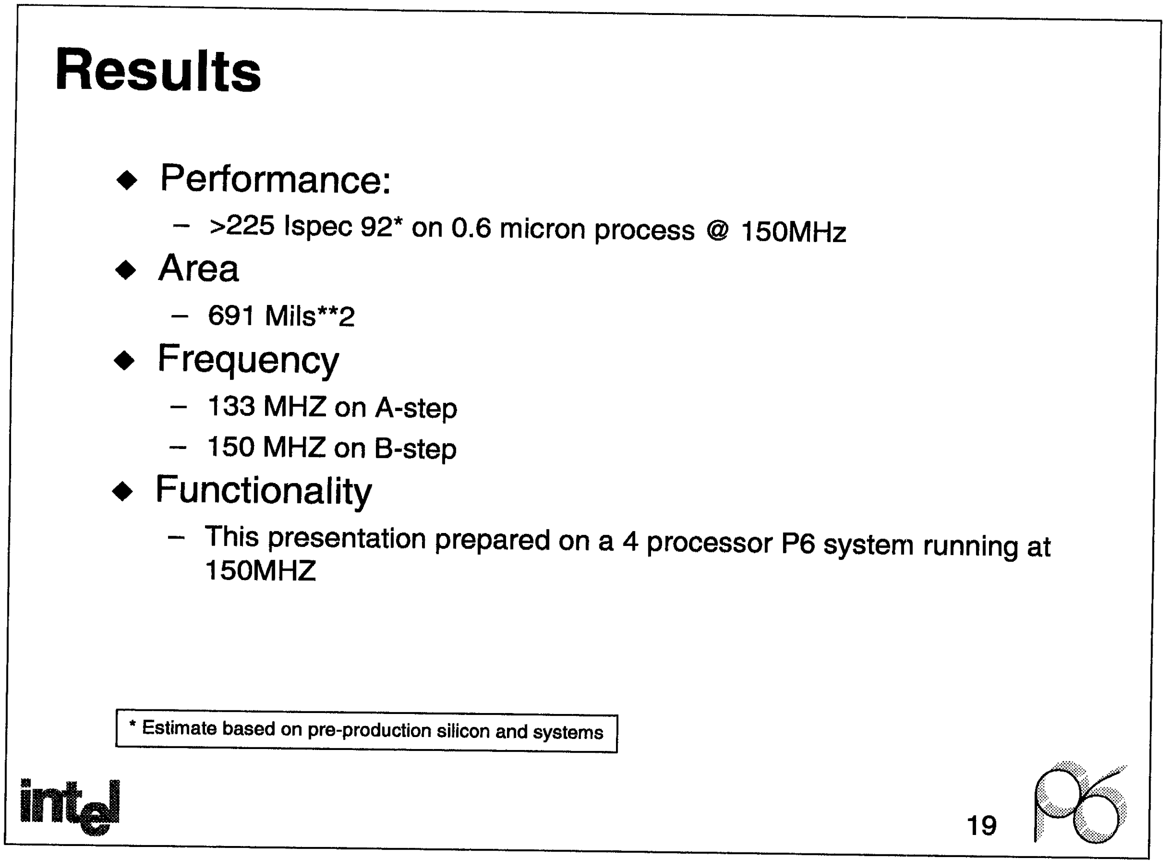
2.1-18



Harder Changes

- 4-1-1 decoder
  - 3% performance loss
- 512 entry BTB
  - 5% loss on TPCB
  - 1-2% on Ispec
- These changes cost more than we liked
  - But needed to hit area and frequency targets
- Careful tuning was critical to success

2.1-19



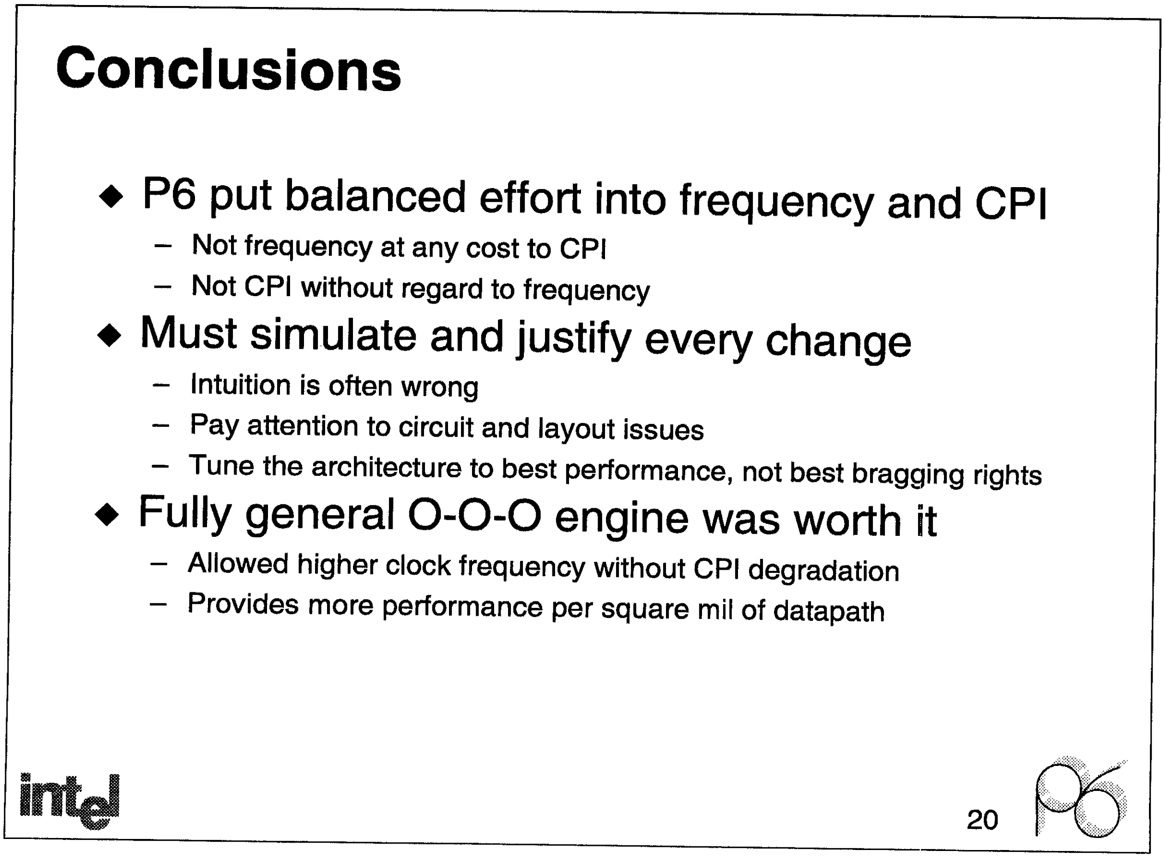
Results

- Performance:
  - >225 Ispec 92* on 0.6 micron process @ 150MHz
- Area
  - 691 Mils**2
- Frequency
  - 133 MHZ on A-step
  - 150 MHZ on B-step
- Functionality
  - This presentation prepared on a 4 processor P6 system running at 150MHZ

* Estimate based on pre-production silicon and systems

19

2.1-20



Conclusions

- P6 put balanced effort into frequency and CPI
  - Not frequency at any cost to CPI
  - Not CPI without regard to frequency
- Must simulate and justify every change
  - Intuition is often wrong
  - Pay attention to circuit and layout issues
  - Tune the architecture to best performance, not best bragging rights
- Fully general O-O-O engine was worth it
  - Allowed higher clock frequency without CPI degradation
  - Provides more performance per square mil of datapath

20