Intel Developer Forum
Fall 2000

Inside the Pentium® 4 Processor Micro-architecture

Next Generation IA-32 Micro-architecture

Doug Carmean
Principal Architect
Intel Architecture Group

August 24, 2000

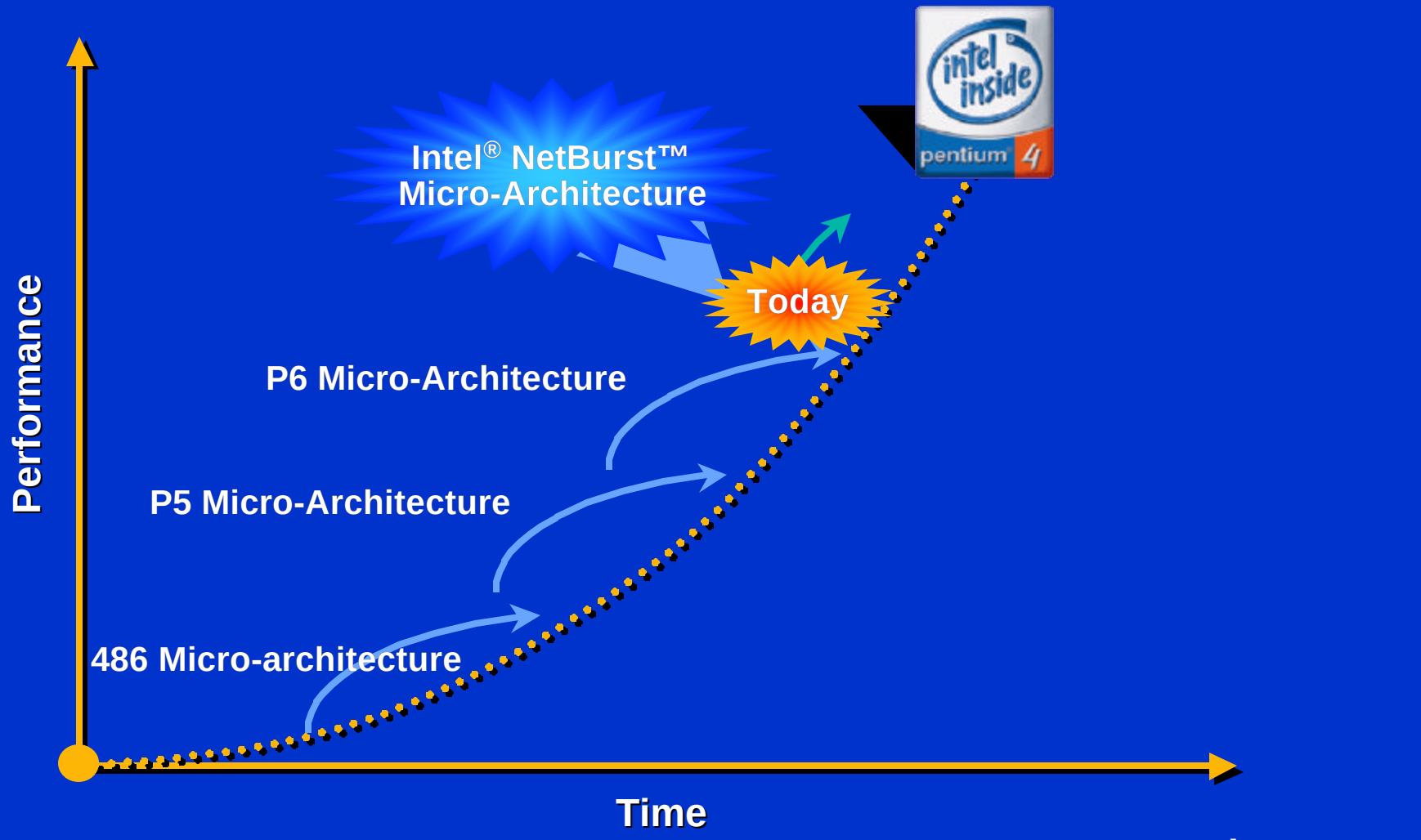
Intel Labs

Agenda

- IA-32 Processor Roadmap
- Design Goals
- Frequency
- Instructions Per Cycle
- Summary

Intel® Pentium® 4 Processor

Intel Developer Forum
Fall 2000



Intel
PDX

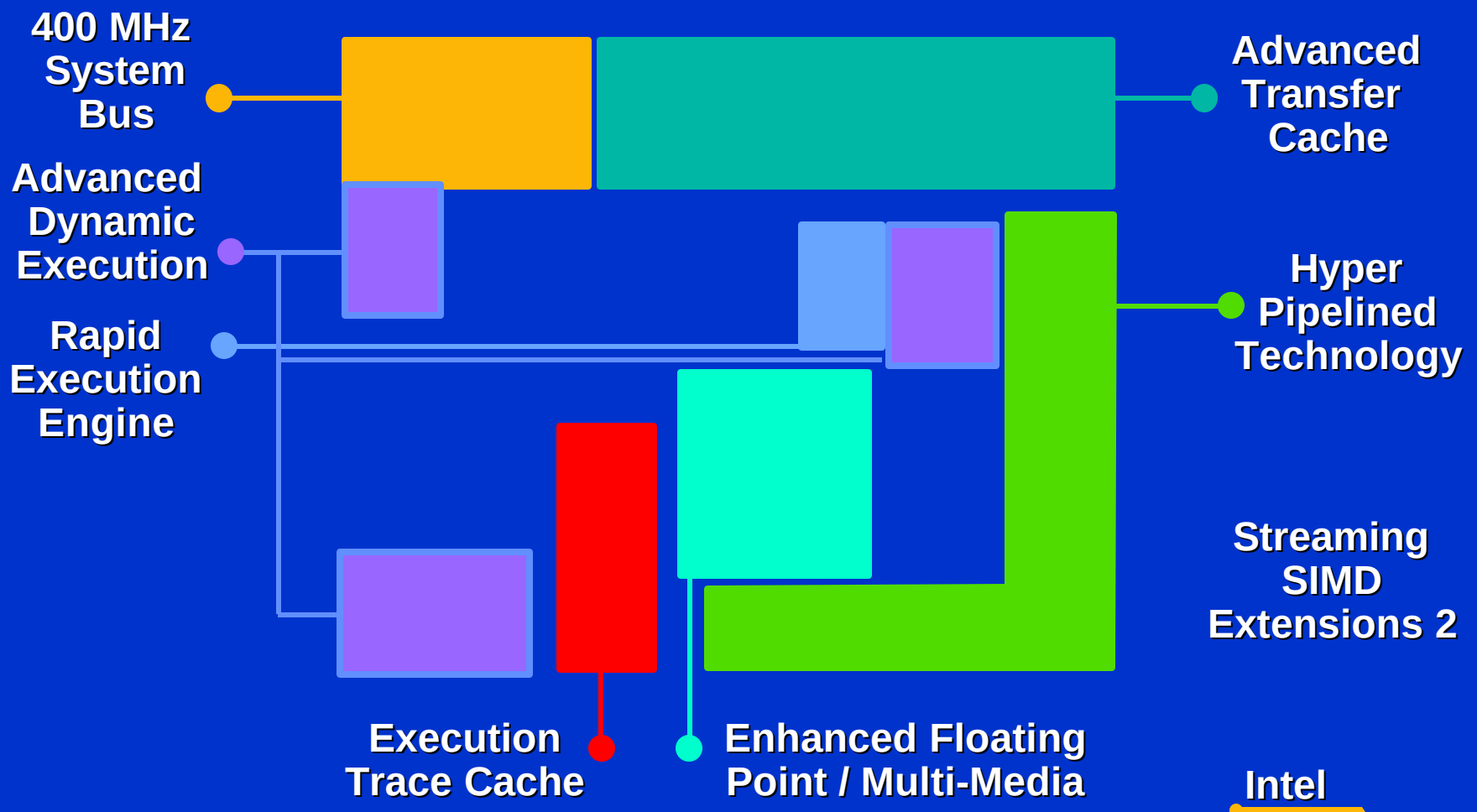
Intel® Pentium® 4 Processor Design Goals

- Deliver world class performance across both existing and emerging applications
- Deliver performance headroom and scalability for the future

Micro-architecture that will Drive Performance Leadership for the Next Several Years

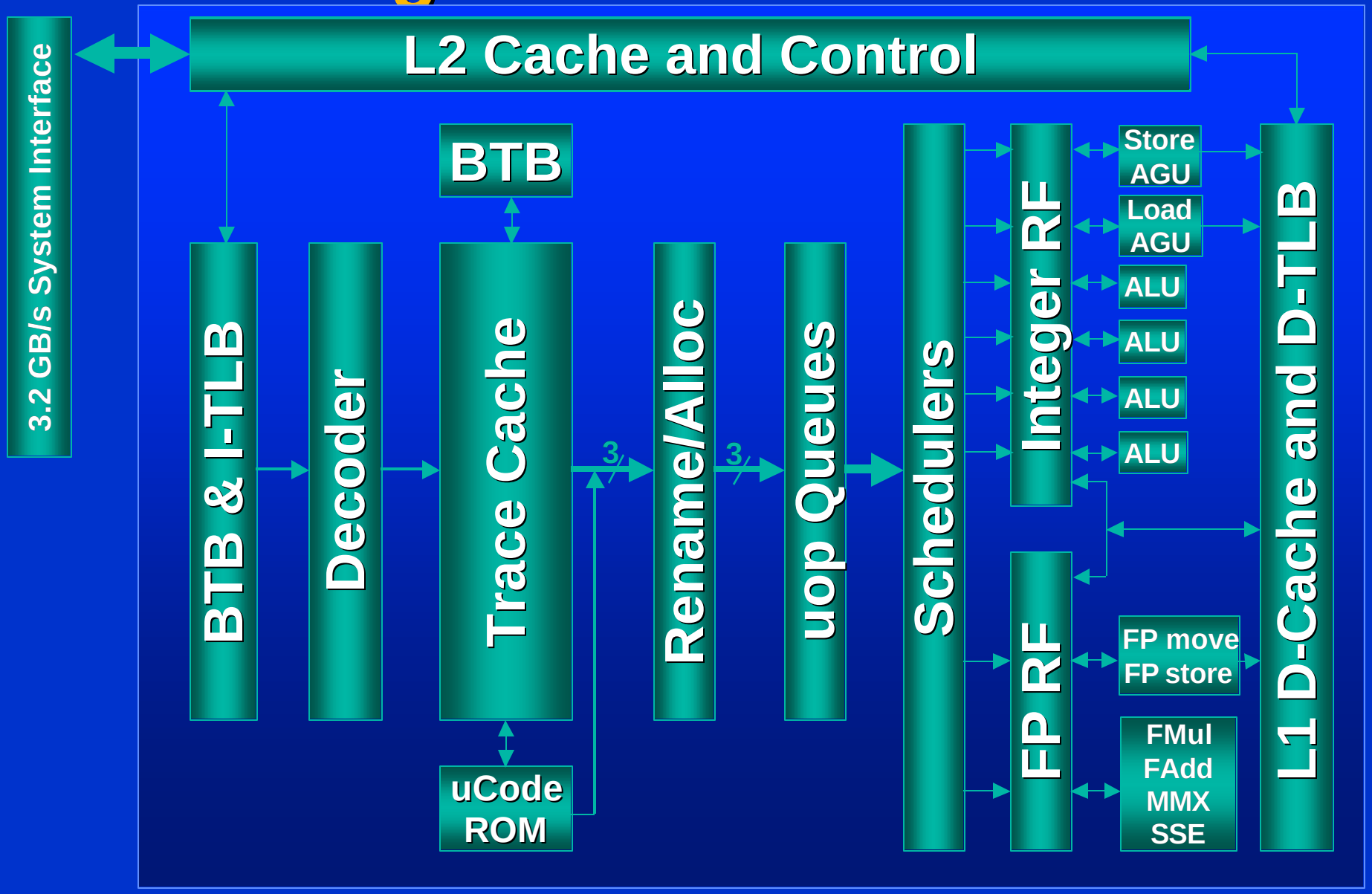
Intel® NetBurst™ Micro-architecture

Intel Developer Forum
Fall 2000

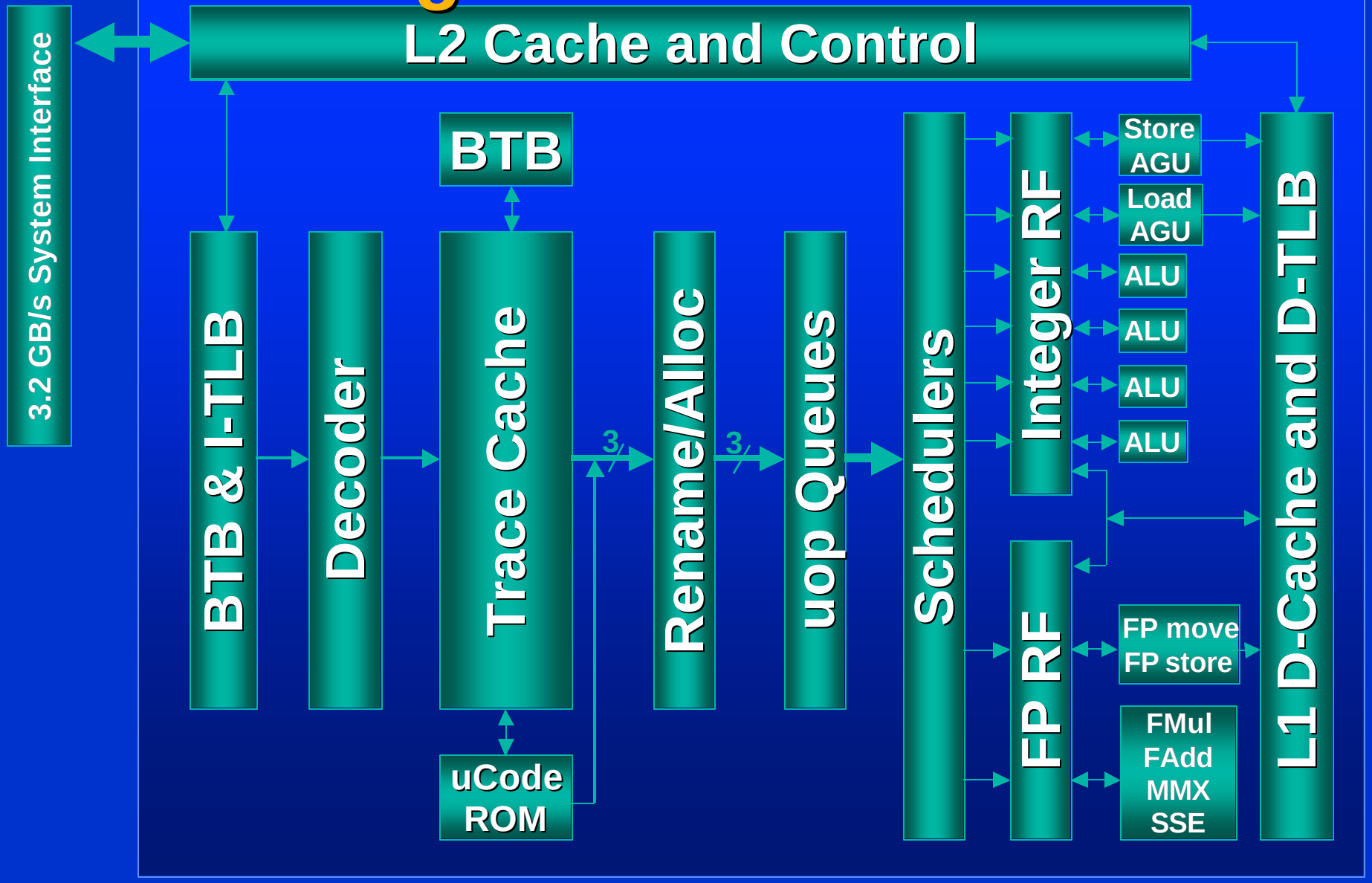


Intel
PDX

Pentium® 4 Processor Block Diagram



Pentium® 4 Processor Block Diagram



CPU Architecture 101

Frequency

- What limits frequency?
  - Process technology
  - Microarchitecture
- On a given process technology
  - Fewer gates per pipeline stage will deliver higher frequency

Frequency is driven by Micro-architecture

Netburst™ Micro-architecture Pipeline vs P6

Intel Developer Forum
Fall 2000

Basic P6 Pipeline

1	2	3	4	5	6	7		.0
Fetch	Fetch	Decode	Decode	Decode	Rename	ROB Rd	Rdy/Sc	Exec

Intro at 733MHz .18µ

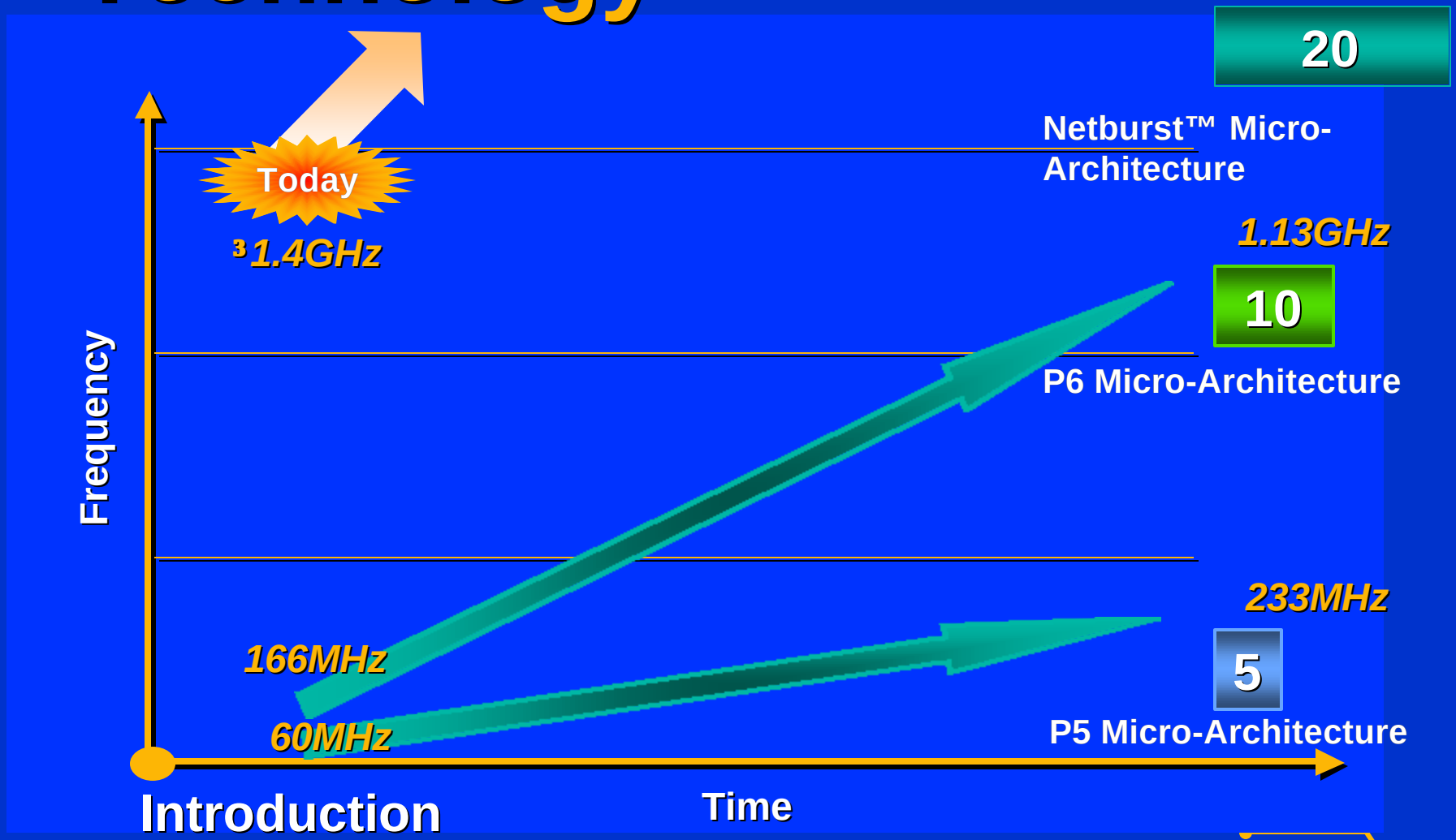
Basic Pentium® 4 Processor Pi

1	2	3	4	5	6	7	8	9	10	11	12	13	14	
TC Nxt IP		TC Fetch		Drive	Alloc	Rename		Que	Sch	Sch	Sch	Disp	Disp	

Intro at 1.4GHz .18µ

Hyper pipelined Technology enables industry leading performance and clock rate

Hyper Pipelined Technology

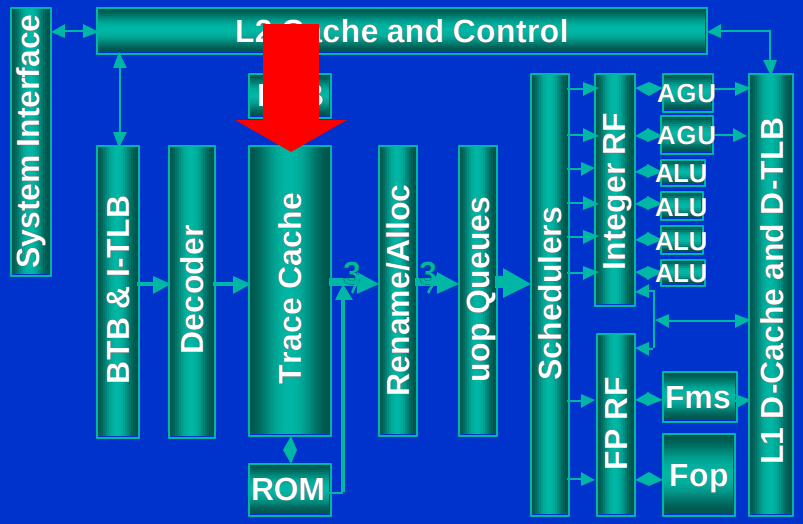


Hyper pipelined Technology

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

TC Nxt IP: Trace cache next instruction pointer

Pointer from the BTB, indicating location of next instruction.

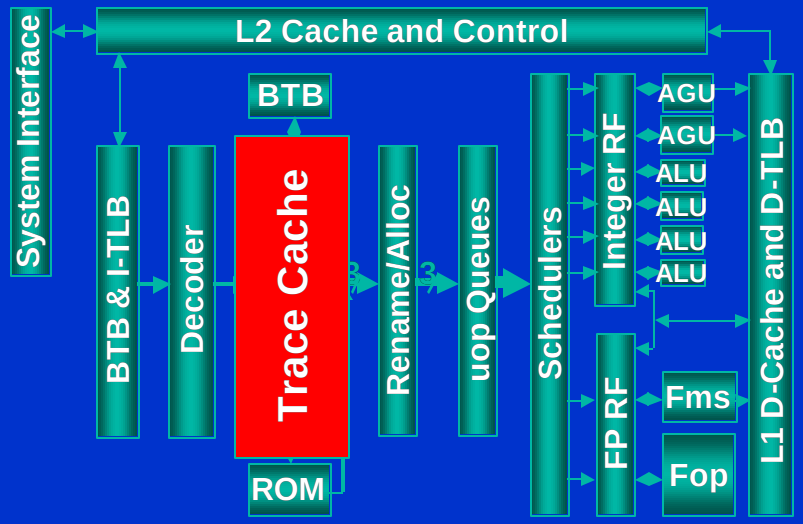


Hyper Pipelined Technology

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

TC Fetch: Trace cache fetch

Read the decoded instructions (uOPs) out of the Execution Trace Cache

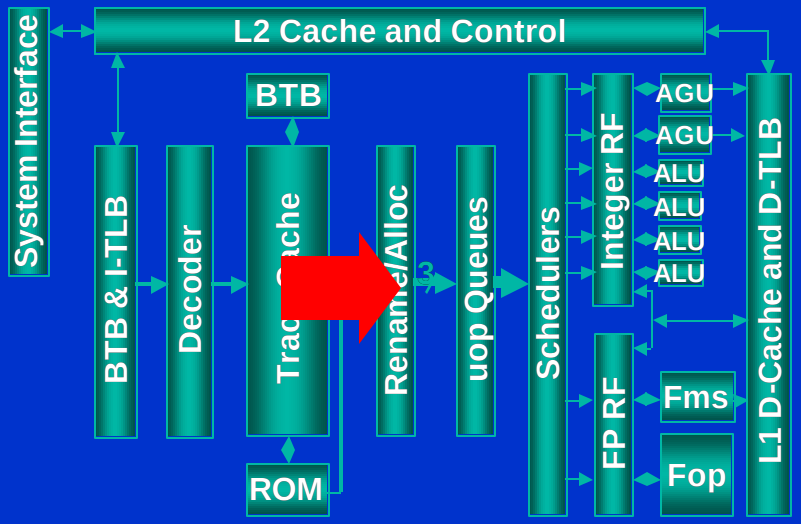


Hyper Pipelined Technology

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

Drive: Wire delay

Drive the uOPs to the allocator

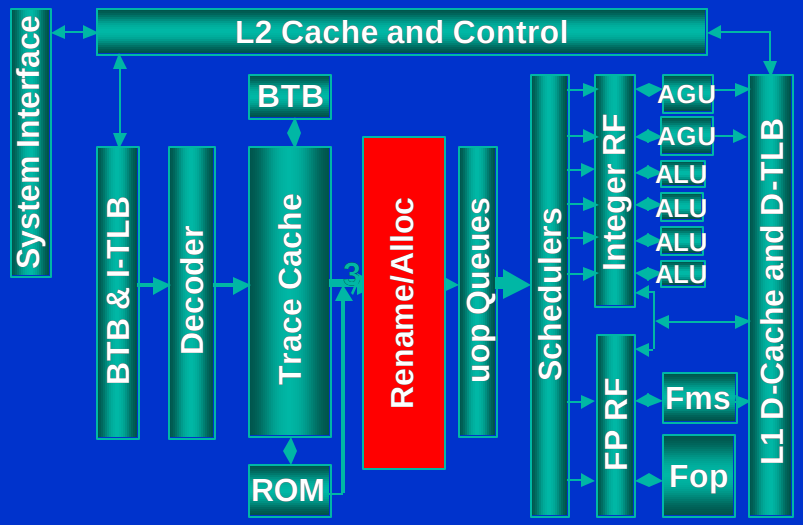


Hyper Pipelined Technology

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

Alloc: Allocate

Allocate resources required for execution. The resources include Load buffers, Store buffers, etc..



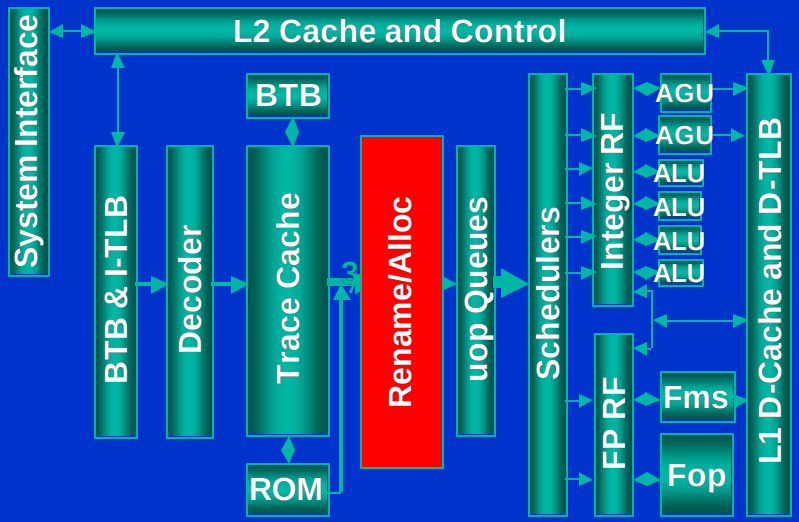
Intel PDX

Hyper Pipelined Technology

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

Rename: Register renaming

Rename the logical registers (EAX) to the physical register space (128 are implemented).



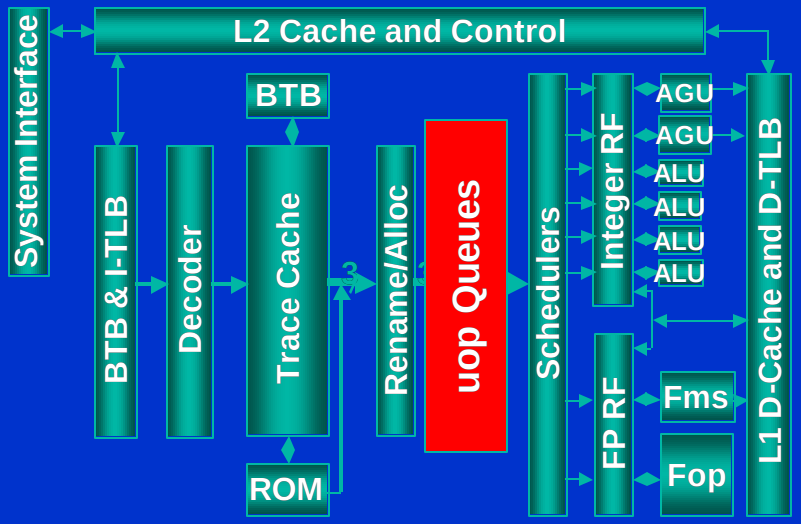
Intel PDX

Hyper Pipelined Technology

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

Que: Write into the uOP Queue

uOPs are placed into the queues, where they are held until there is room in the schedulers

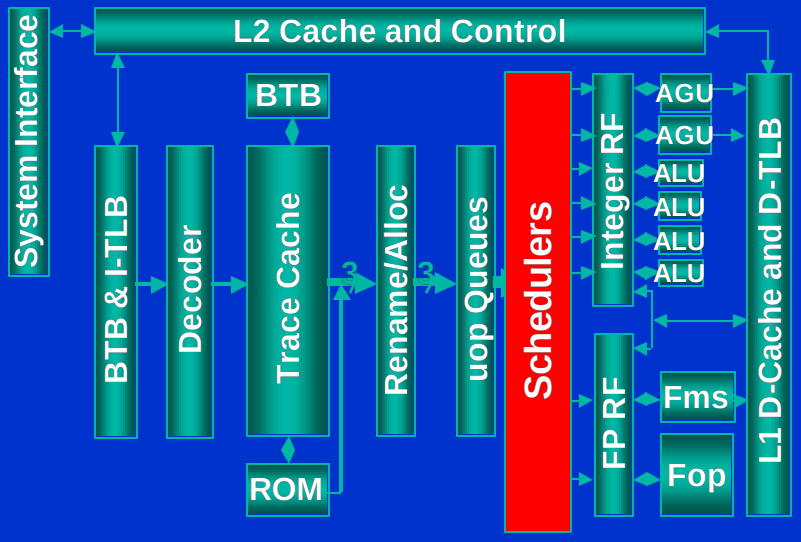


Hyper Pipelined Technology

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

Sch: Schedule

Write into the schedulers and compute dependencies. Watch for dependency to resolve.

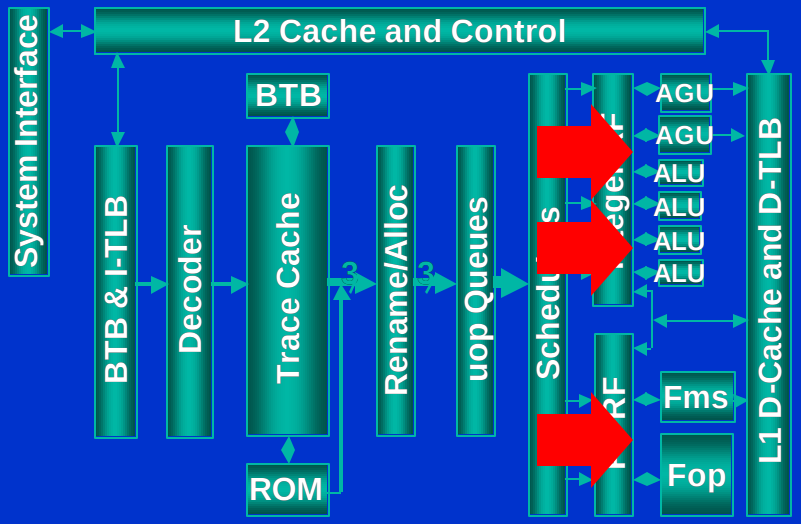


Hyper Pipelined Technology

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

Disp: Dispatch

Send the uOPs to the appropriate execution unit.

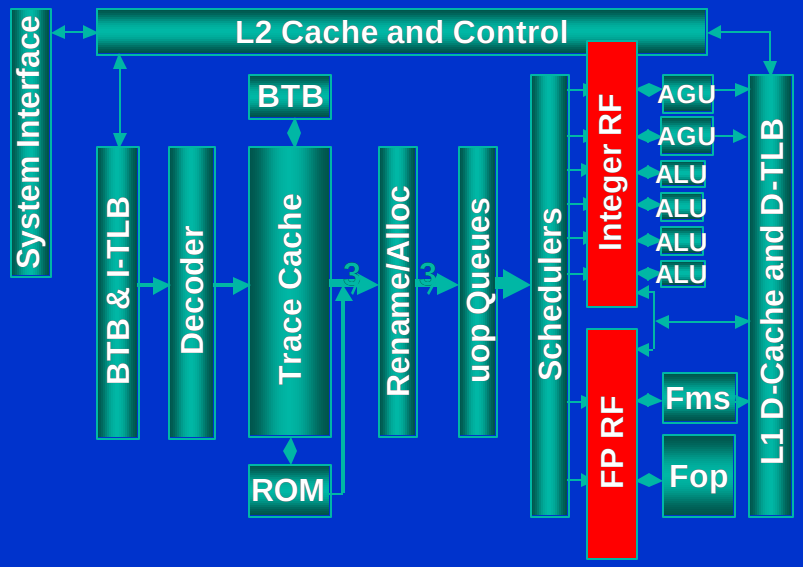


Hyper Pipelined Technology

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

RF: Register File

Read the register file. These are the source(s) for the pending operation (ALU or other).

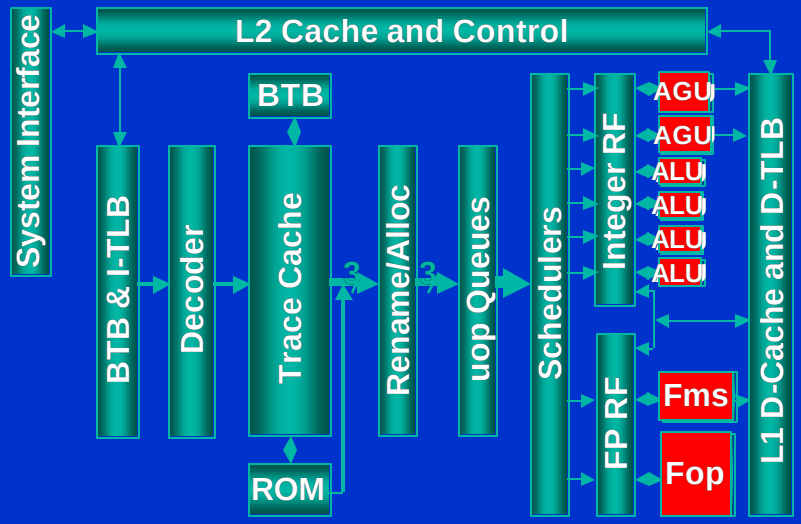


Hyper Pipelined Technology

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

Ex: Execute

Execute the uOPs on the appropriate execution port.

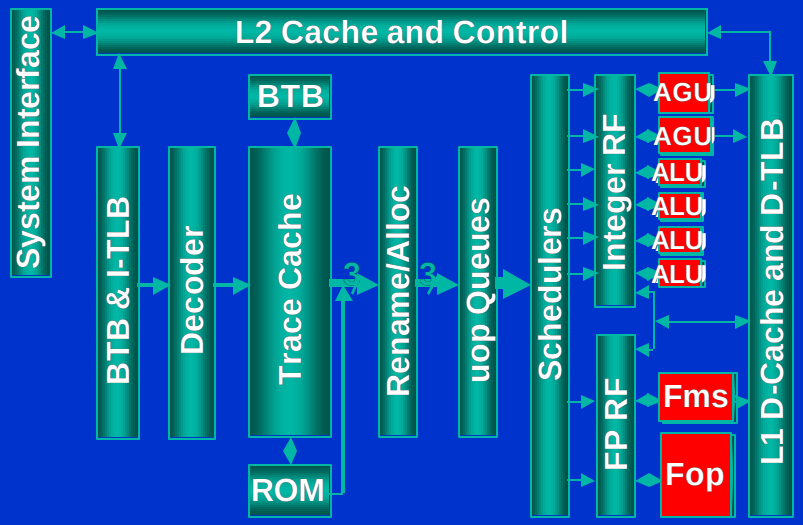


Hyper Pipelined Technology

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

Flgs: Flags

Compute flags (zero, negative, etc..). These are typically the input to a branch instruction.

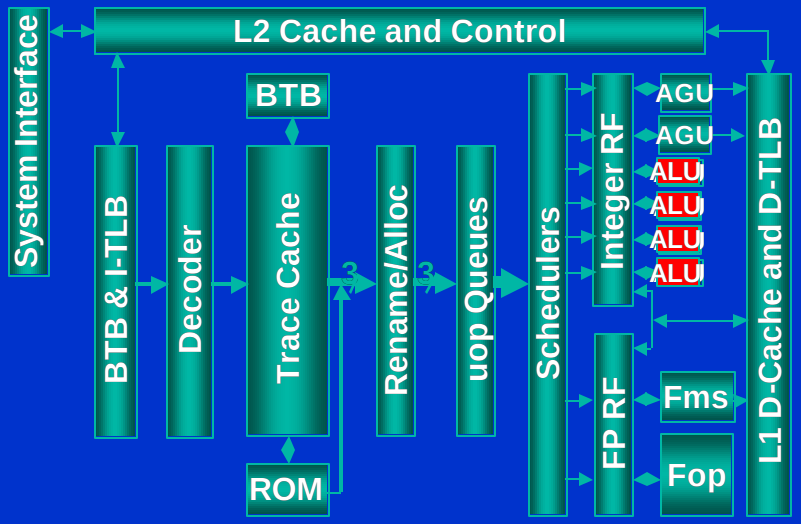


Hyper Pipelined Technology

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

Br Ck: Branch Check

The branch operation compares the result of the actual branch direction with the prediction.

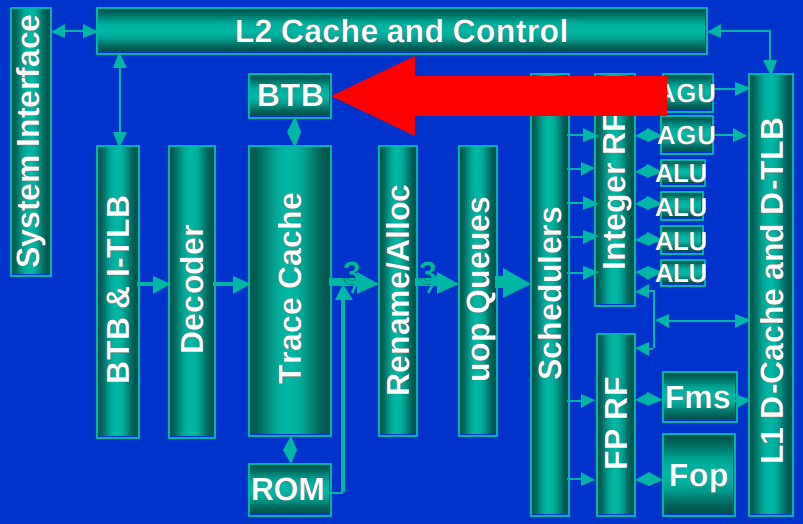


Hyper Pipelined Technology

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

Drive: Wire delay

Drive the result of the branch check to the front end of the machine.



Intel PDX

CPU Architecture 101

Improving Instructions Per Cycle

- Improve efficiency
  - Do more things in a clock
  - Branch prediction
- Reduce time it takes to do something
  - Reducing latency

Improving Instructions Per Cycle

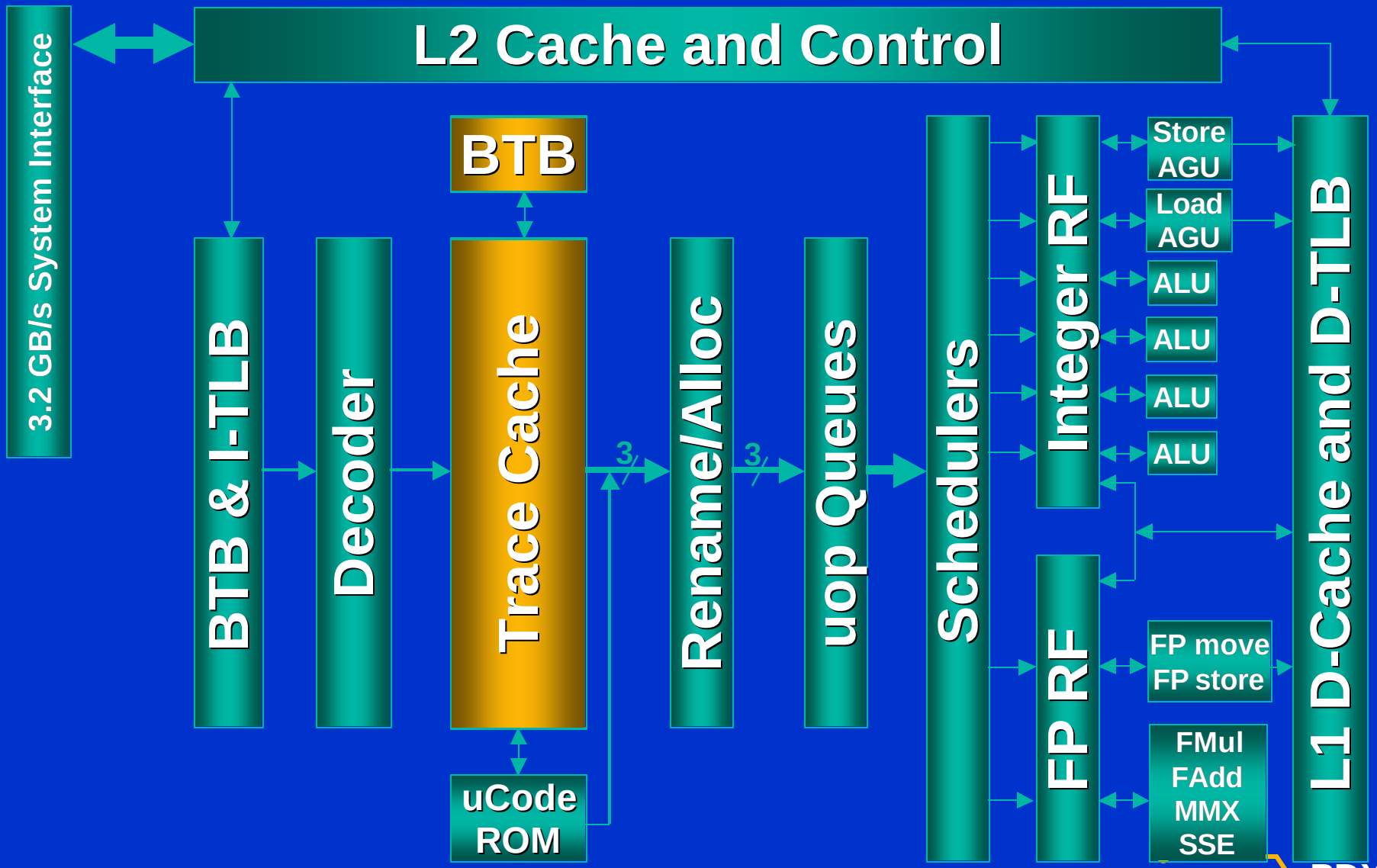
- Improve efficiency
  - Branch prediction
  - Do more things in a clock
- Reduce time it takes to do something
  - Reducing latency

Branch Prediction

- Accurate branch prediction is key to enabling longer pipelines
- Dramatic improvement over P6 branch predictor:
  - 8x the size (4K)
  - Eliminated 1/3 of the mispredictions
- Proven to be better than *all* other publicly disclosed predictors
  - (g-share, hybrid, etc)

The Execution Trace Cache

Intel Developer Forum
Fall 2000

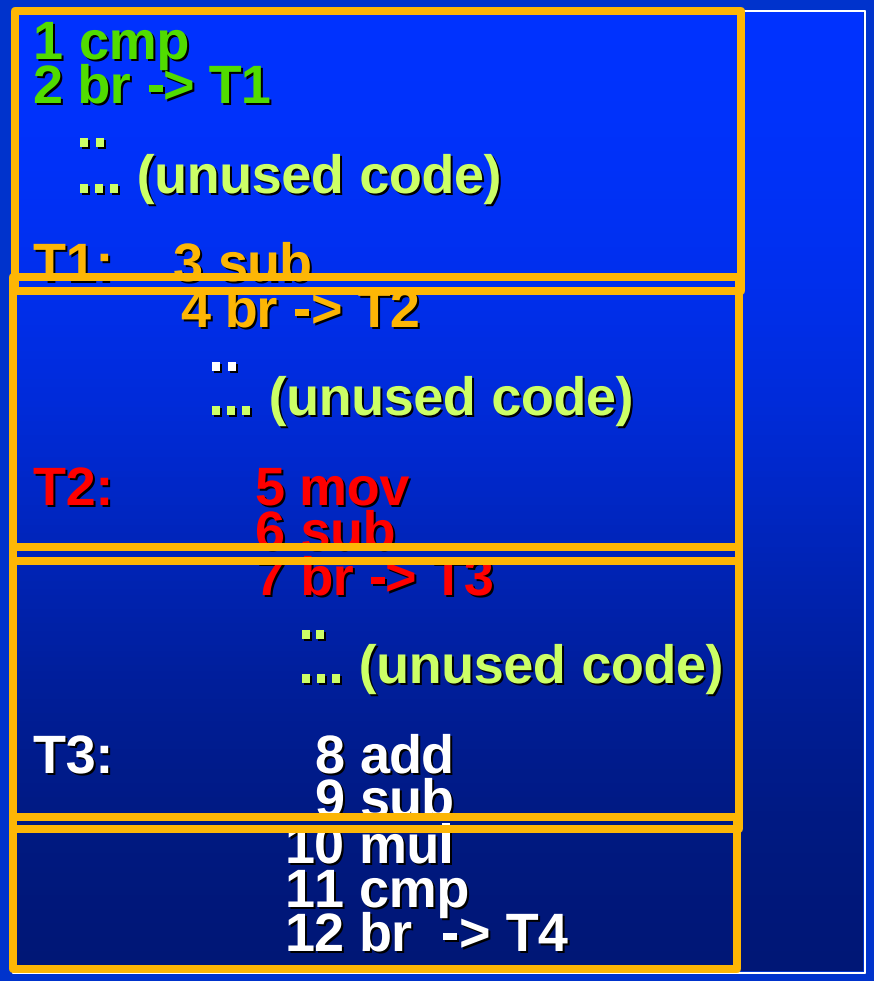


Execution Trace Cache

- Advanced L1 instruction cache
  - Caches “decoded” IA-32 instructions (uops)
- Removes decoder pipeline latency
- Capacity is ~12K uOps
- Integrates branches into single line
  - Follows predicted path of program execution

Execution Trace Cache feeds fast engine

Execution Trace Cache



Trace Cache Delivery

1 cmp	2 br T1	3 T1: sub
4 br T2	5 mov	6 sub
7 br T3	8 T3:add	9 sub
10 mul	11 cmp	12 br T4

Advanced Dynamic Execution

- Extends basic features found in P6 core
- Very deep speculative execution
  - 126 instructions in flight (3x P6)
  - 48 loads (3x P6) and 24 stores (2x P6)
- Provides larger window of visibility
  - Better use of execution resources

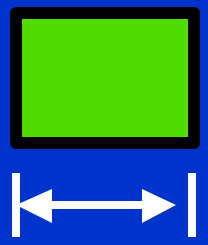
Deep Speculation Improves Parallelism

Improving Instructions Per Cycle

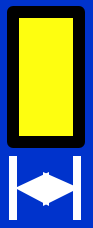
- Improve efficiency
  - Do more things in a clock
  - Branch prediction
- Reduce time it takes to do something
  - Reducing latency

Rapid Execution Engine

- Dramatically lower ALU latency
- P6:
  - 1 clock @ 1GHz

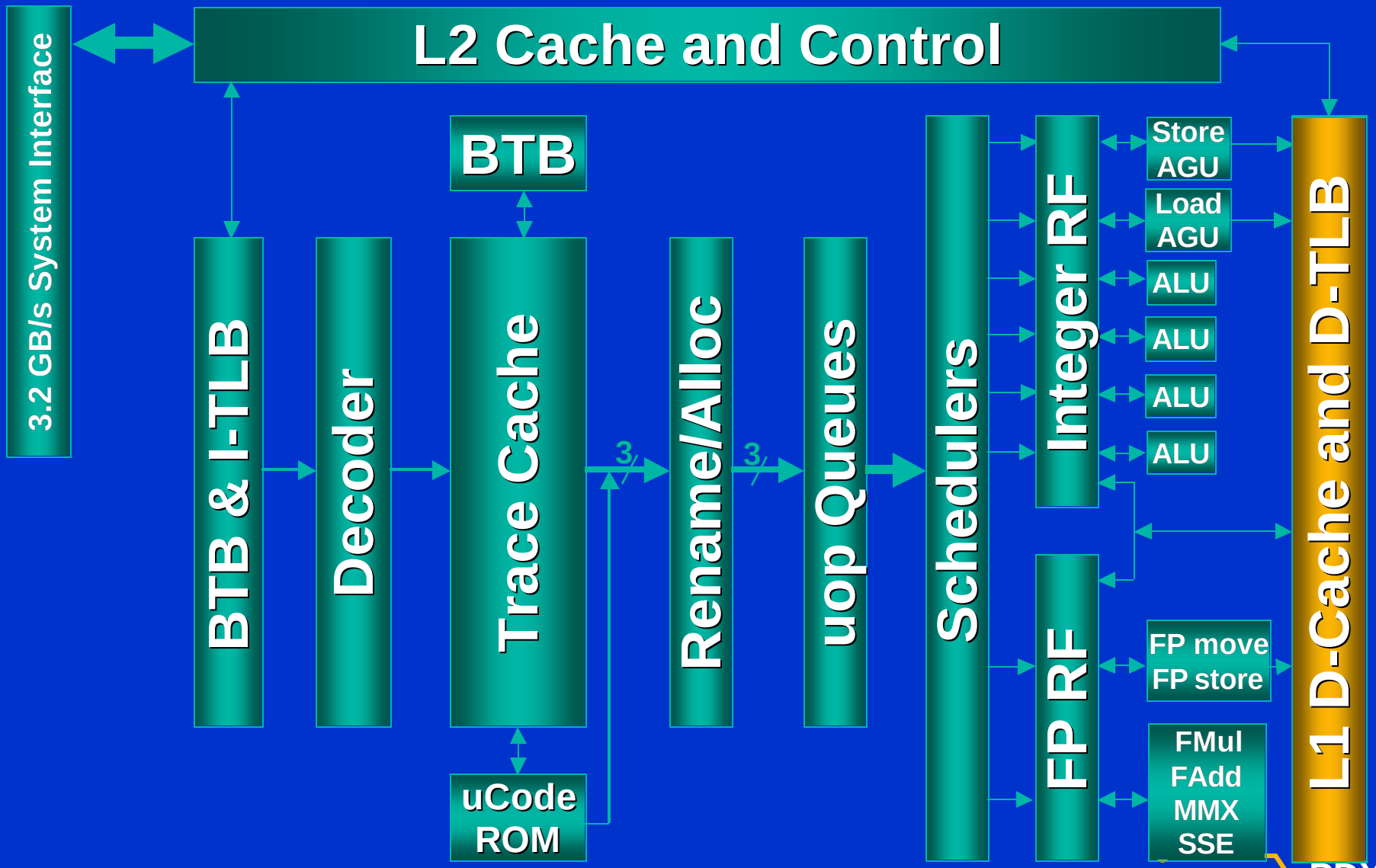


- Intel® NetBurst™ micro-architecture:
  - ½ clock @ >1.4GHz



L1 Data Cache

Intel Developer Forum
Fall 2000



High Performance L1 Data Cache

- 8KB, 4-way set associative, 64-byte lines
- Very high bandwidth
  - 1 Ld and 1 St per clock
- New access algorithms
- Very low latency
  - 2 clock read

New algorithm enables faster cache

Data Speculation

- Observation: Almost all memory accesses hit in the cache
- Optimize for the common case
  - Assume that the access will hit the cache
  - Use a low cost mechanism to fix the rare cases that miss
- Benefit:
  - Reduces latency
  - Significantly higher performance

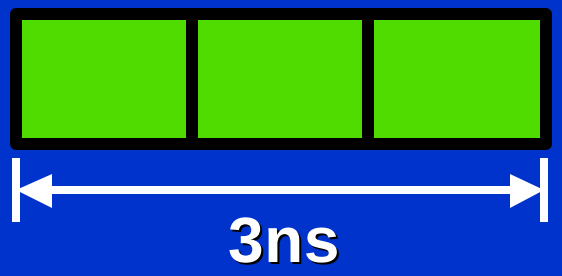
Replay

- Repairs incorrect speculation
  - Re-execute until correct
- Replay is uOP specific
  - Replay the uOP that mis-speculated
  - Replay dependent uOPs
  - Independent uOPs are not replayed

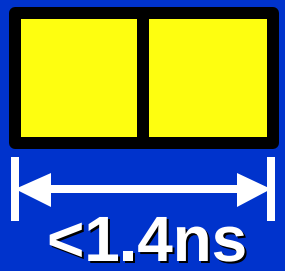
Efficient mechanism to reduce latency

L1 Cache is >2x Faster

- P6:
  - 3 clocks @ 1GHz



- Intel® NetBurst™ micro-architecture:
  - 2 clocks @ ³1.4GHz



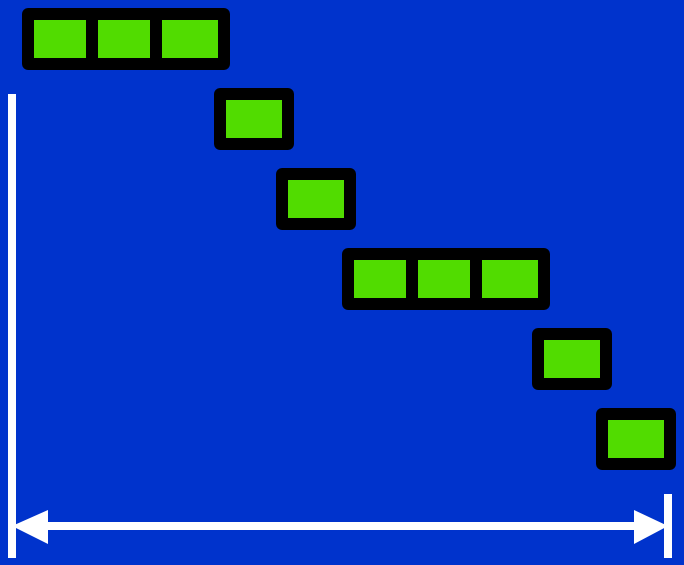
Lower Latency is Higher Performance

Example with higher IPC and Faster Clock!

Code Sequence

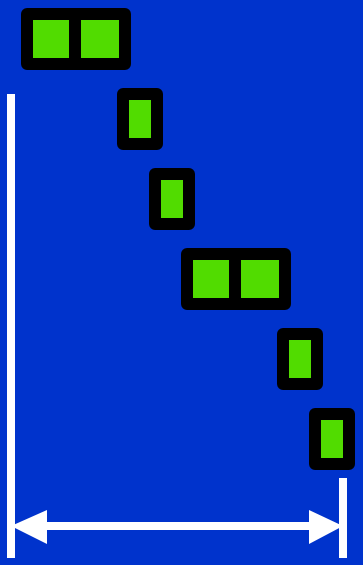
```
Ld
Add
Add
Ld
Add
Add
```

P6 @1GHz



10 clocks
10ns
IPC = 0.6

Intel® NetBurst™ Micro-architecture @1.4GHz

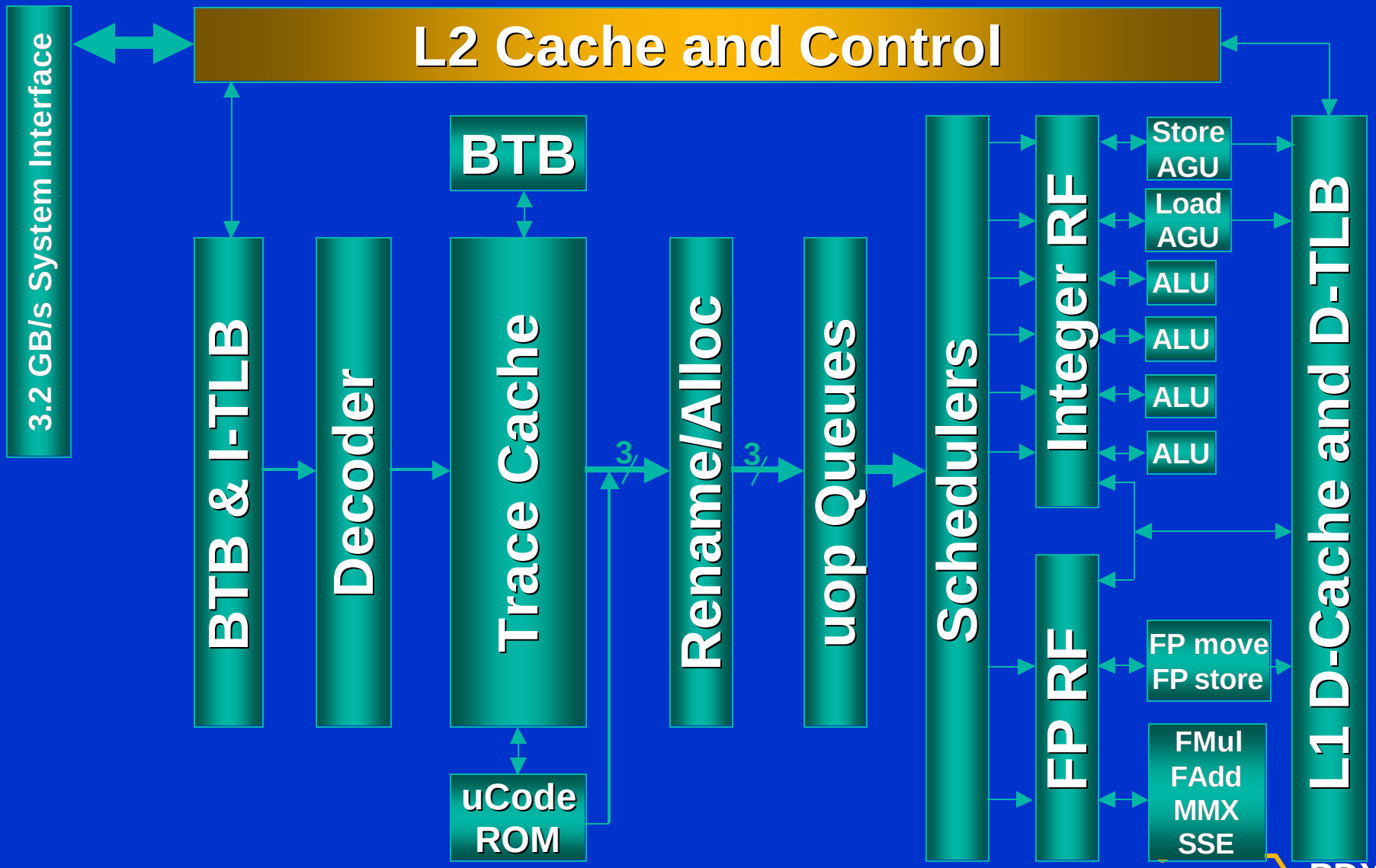


6 clocks
4.3ns
IPC = 1.0

PDX

L2 Advanced Transfer Cache

Intel Developer Forum
Fall 2000



L2 ATC Organization

- 256KB, 8-way set associative
  - 128-byte lines
  - Two 64-byte pieces per line
- Holds both data and instructions
- High bandwidth: 45 GB/Sec @ 1.4GHz
  - 2.8x P6 @1GHz

Aggregate Cache Latency

- Function of all caches in a processor
- Overall Effective Latency

  L1 latency +

  L1 Miss Rate * L2 latency +

  L2 Miss Rate * DRAM Latency

Average cache speed is >1.8x better than the Pentium® III Processor

Average on desktop applications,
Intel® Pentium® III processor @ 1GHz, Intel® Pentium® 4 processor @ 1.4GHz

Comparison

	Pentium® III Processor	Pentium 4 Processor	Relative Improvement
Frequency	1 GHz	³ 1.4 Ghz	³ 1.4
Adder Speed	1 ns	< .36 ns	³ 2.8
Adder Bandwidth	2 billion/sec	³ 5.6 billion/sec	> 2.8
L1 Cache Speed	3 ns	< 1.42 ns	³ 2.1
L1 Cache Size	16 KB	8 KB	0.5
L1 Cache Bandwidth	16 GB/sec	³ 44.8 GB/sec	³ 2.8
L2 Cache Bandwidth	16 GB/sec	³ 44.8 GB/sec	³ 2.8
Instructions In flight	40	126	3.15
Loads in flight	16	48	3
Stores in flight	12	24	2
Branch targets	512	4092	8
Uop Fetch Bandwidth	3 billion/sec	³ 4.2 billion/sec	³ 1.4

Intel® Pentium® 4 Processor Summary

- Revolutionary, new micro-architecture from Intel designed for the evolving Internet
- Design features for balanced, high performance platform scalability and headroom
- The world's *highest performance* desktop processor