System Architecture

PC Architecture
Part II

Ashish Kuvelkar

HPC Technologies Group

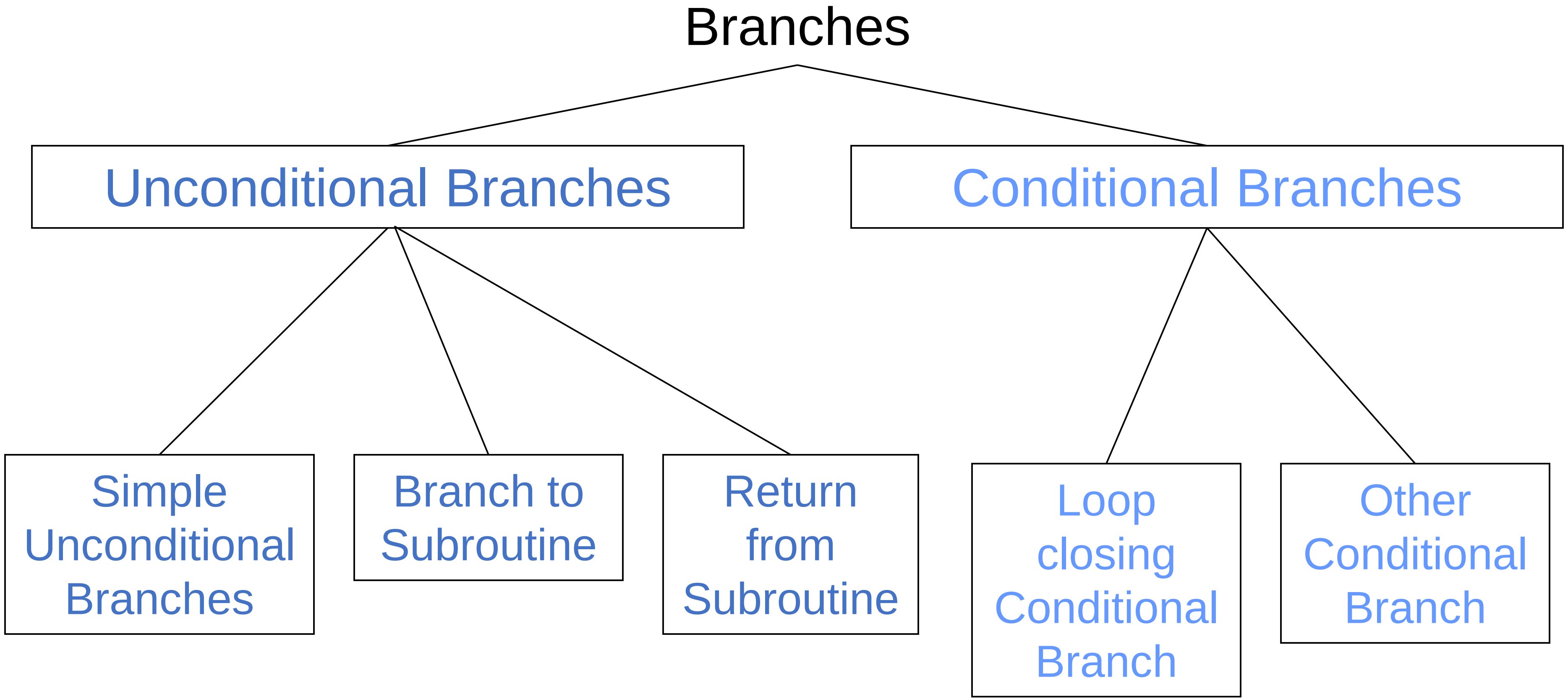
C-DAC, Pune

Processing of Control Transfer Instructions

Branch Statistics

- Branches account for 20% of General purpose code
  - 5-10% of scientific code
- Conditional vs. non-conditional
  - One-third are unconditional
  - One-third are conditional loop closing
  - One-third are other conditional
- Taken vs. untaken branches
  - Taken are ~5/6
  - Untaken are 1/6

Types of Branches


Branches

Unconditional Branches

Conditional Branches

Simple Unconditional Branches

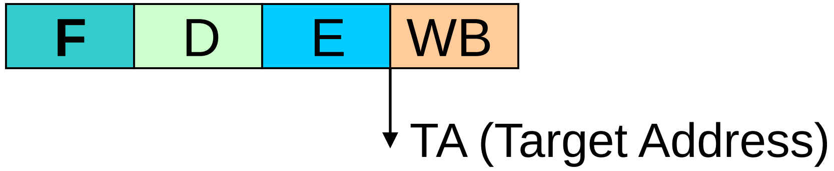
Branch to Subroutine

Return from Subroutine

Loop closing Conditional Branch

Other Conditional Branch

The branch problem



	t_i	t_{i+1}	t_{i+2}	t_{i+3}	t_{i+4}	
i_1	F	D	E	WB		
b		F	D	E		
i_3			~~F~~	~~D~~	T	
i_4				~~F~~	A	
i_{t1}					F	

Delay of 2 cycles

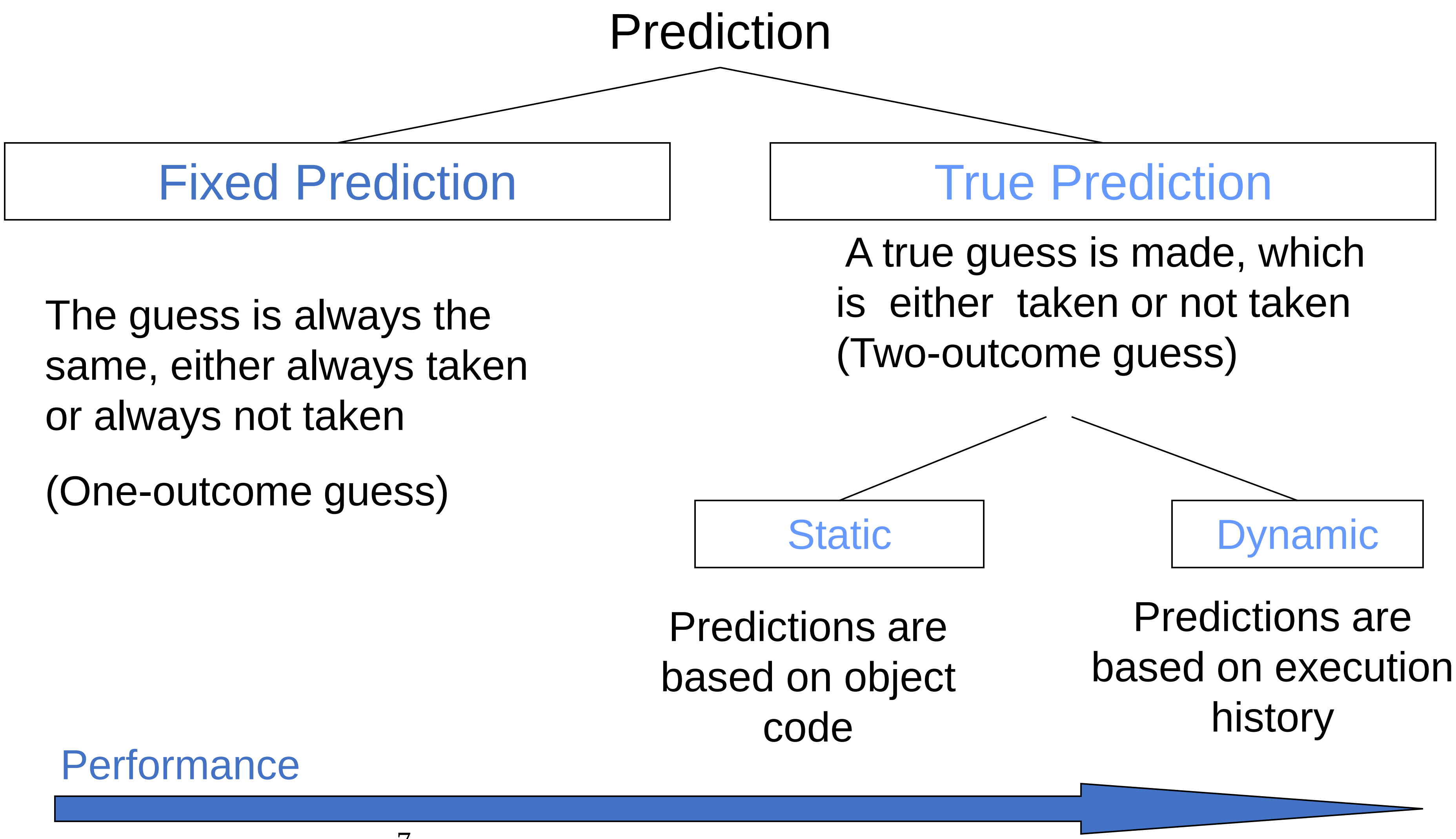
```
         i1
         b    target
Nochng:i3
         i4
         .
         .
target:it1
          it2
```

The branch problem

- Unconditional branch causes 2 cycle delay
- Conditional branch could cause even longer penalty
- Situation is worse for unresolved conditional branch
    - Result referred in condition is not yet produced
        - E.g. floating point division (10-50 cycles)
- In a pipeline, each branch gives rise to a number of wasted cycles (called **bubbles**)
- Pentium has 8 stage pipeline, pentium pro has 12
- Super-scaler processors issue multiple instructions in a cycle

Efficient Branch processing helps in improving performance

Branch Prediction Scheme



Prediction

Fixed Prediction

The guess is always the same, either always taken or always not taken

(One-outcome guess)

True Prediction

A true guess is made, which is either taken or not taken (Two-outcome guess)

Static

Predictions are based on object code

Dynamic

Predictions are based on execution history

Performance

Fixed Prediction

Always Not Taken

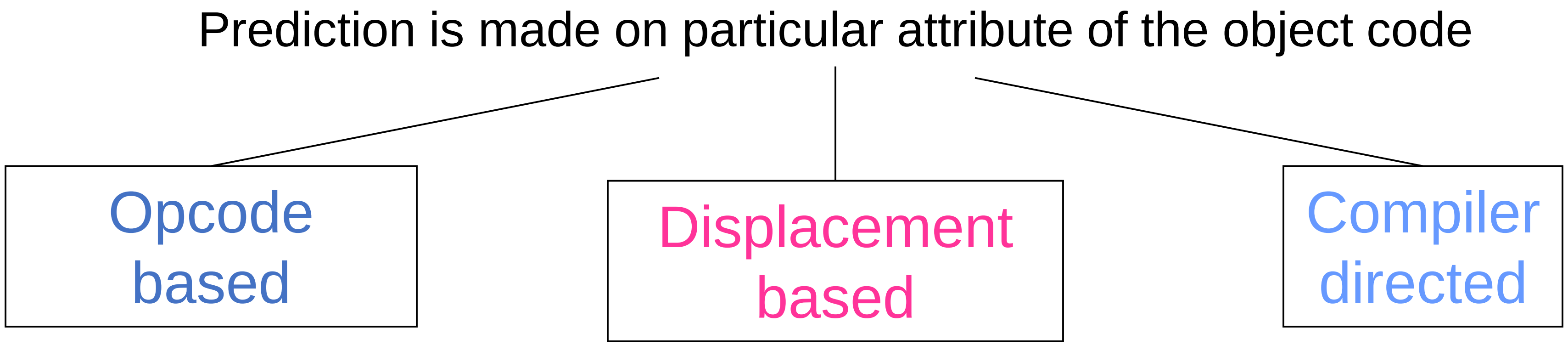
- Always Guess an unresolved branch as ‘not taken’
- Continue execution of sequential path
- In preparation for wrong guess, start execution of taken path in parallel (e.g calculate BTA)
- When the condition becomes evaluable, check guess
- If guess correct, continue with execution of sequential path and delete taken path preprocessing
- If the guess is incorrect delete all speculative by executed instructions and continue with taken path processing

Always Taken

- Always Guess an unresolved branch as ‘ taken’
- In anticipation of wrong guess, save processing status (e.g the PC)
- When the condition becomes available, check guess
- If guess correct, continue with execution of taken path and delete the saved status
- If the guess is incorrect delete the speculative processing and continue with execution of sequential path using saved processing status

True Prediction

Static Prediction



Prediction is made on particular attribute of the object code

Opcode based

Displacement based

Compiler directed

For certain opcodes the branch is assumed as taken, for others as not taken

If D< 0, taken

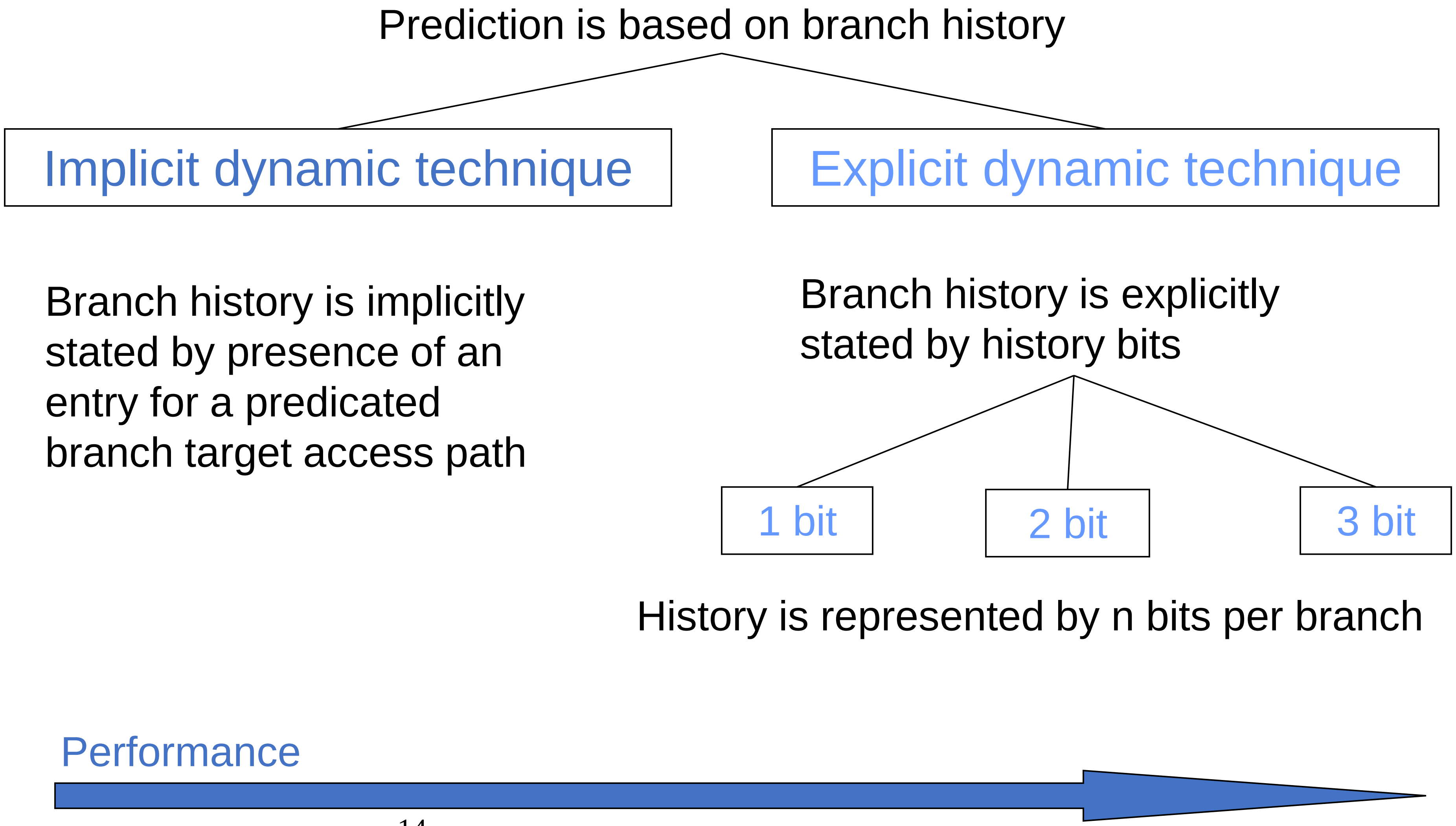
If D>= 0, not taken

Based on kind of control construct compiled, the predict bit is set or reset and encoded in branch instruction

Dynamic Prediction

- Based on branch history
  - Branches that were taken in last n occurrences are also likely to be taken in the next occurrene
- Have higher performance potential than static ones
  - Implementation in complex
  - Storage and updation of pervious outcomes of a large number of branches need to be done
- Explicit: stated explicitly by means of history bits
- Implicit: stated implicitly by presence of an entry for a predicted branch target access path

Dynamic Prediction



Prediction is based on branch history

Implicit dynamic technique

Explicit dynamic technique

Branch history is implicitly stated by presence of an entry for a predicated branch target access path

Branch history is explicitly stated by history bits

1 bit

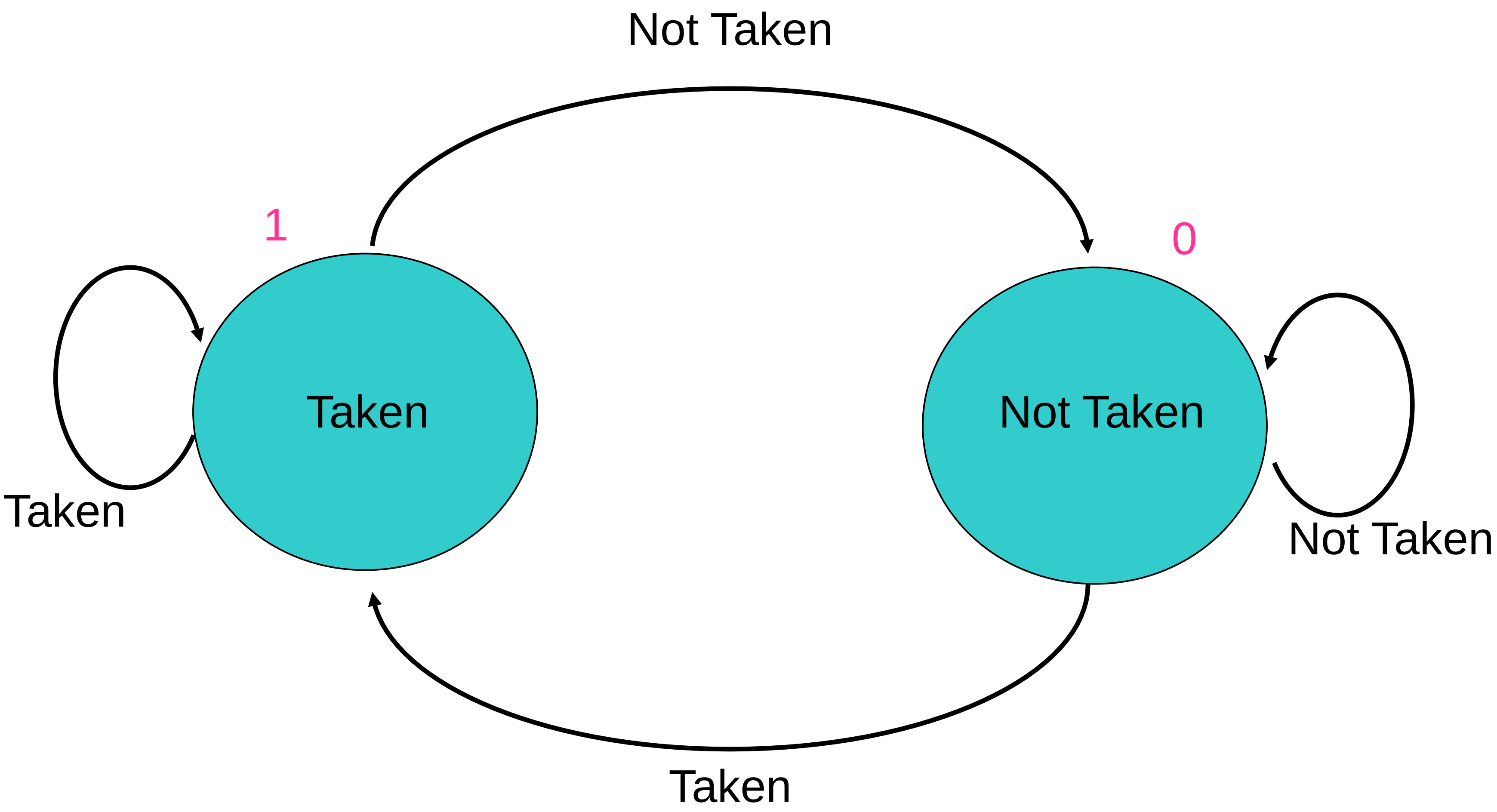
2 bit

3 bit

History is represented by n bits per branch

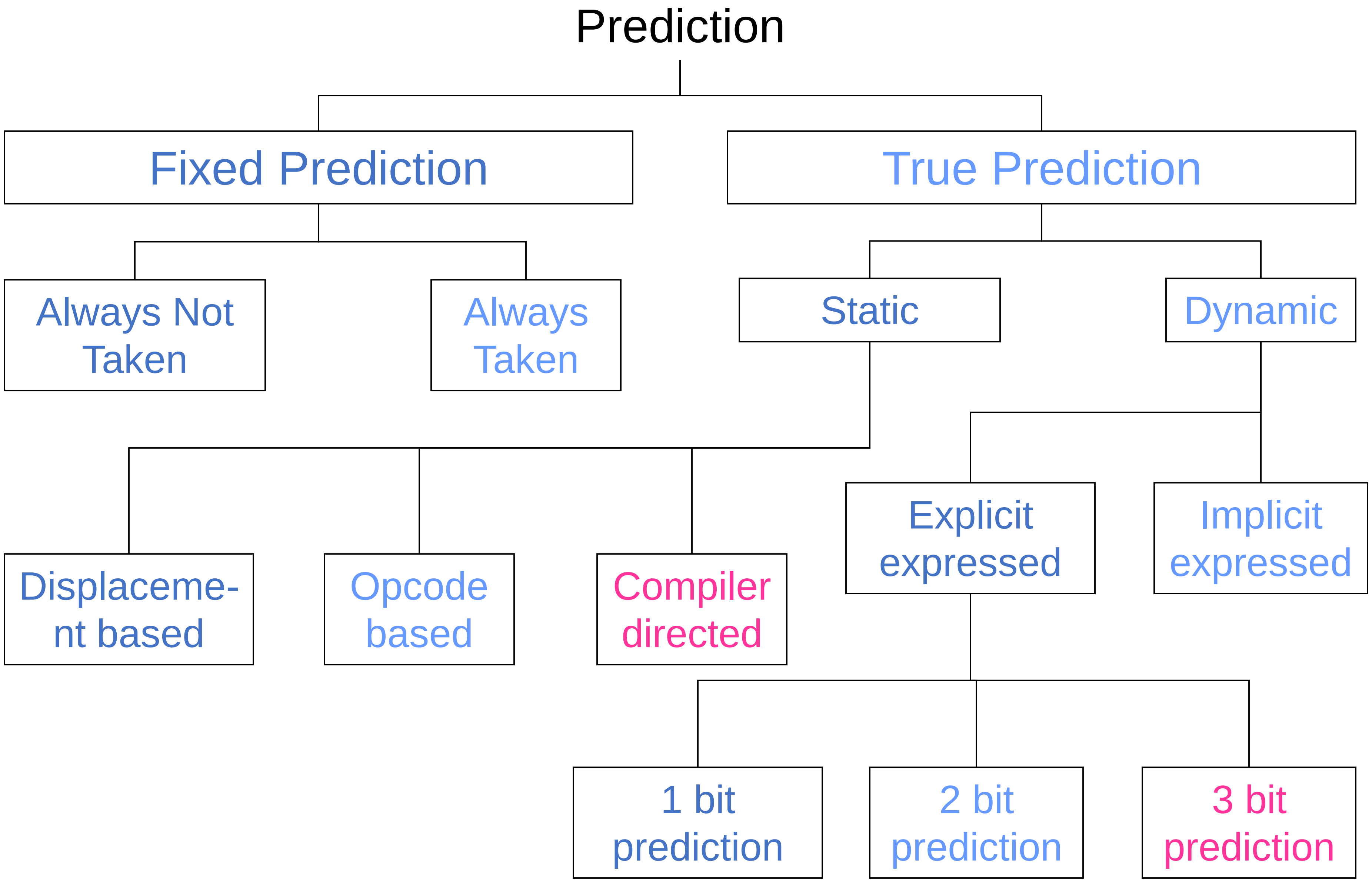
Performance

1 bit dynamic prediction



Implicit Dynamic Technique

- Uses Branch Target Path access schemes
  - BTAC (Branch Target Access Cache)
  - BTIC (Branch Target Instruction Cache)
- It has extra cache which holds either corresponding branch target address or branch target instruction
- Entries in BTAC or BTIC are held only for taken branches
- Branch history is implicitly indicated by presence or absence of entry.
- Existence of entry indicates branch was taken last time, so next occurrence is also guessed as ‘taken’

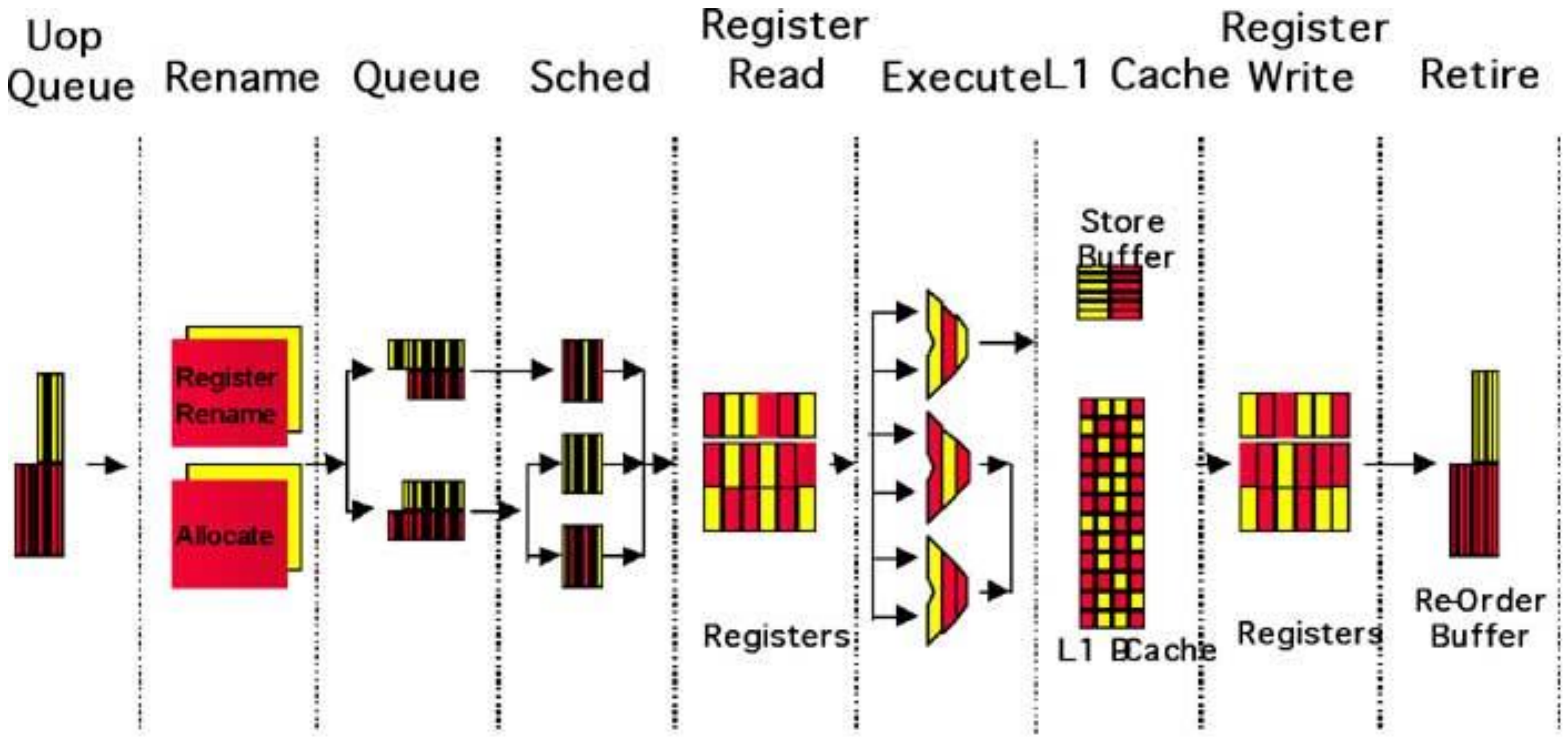


- Prediction
  - Fixed Prediction
    - Always Not Taken
    - Always Taken
  - True Prediction
    - Static
      - Displaceme-nt based
      - Opcode based
      - Compiler directed
    - Dynamic
      - Explicit expressed
        - 1 bit prediction
        - 2 bit prediction
        - 3 bit prediction
      - Implicit expressed

Pentium Pro

- RISC translator within a CISC shell ??
- 1995, 5.5 Million transistors, 166, 200 MHz
- Multi-chip module: CPU core + 256 KB L1 cache
  - L1 cache gives very good performance, since PCB related delays are eliminated
  - Most applications require < 1 MB cache
- ‘RISC Ops’ (rops)
  - 80x86 instructions => Rops => execution => results
  - This works well for 32-bit app (native mode)
- Out of order execution, Register renaming

Pentium



Uop Queue
Rename
Queue
Sched
Register Read
Execute
L1 Cache
Register Write
Retire
Register Rename
Allocate
Registers
Store Buffer
L1 Cache
Registers
Re-Order Buffer

Pentium II

- 1997, 7.5M transistors, 0.28μ, 300,350,450 MHz
- CPU and cache in ‘slot-1’ package
  - licensing issues
- Separate data path for cache memories
- Higher external bus speeds(100MHz)
- Higher CPU clocks (450 MHz)
- Variants based on clock speed and cache memory

Enhancements in architecture

- MMX extensions
  - Performance improvement of multimedia and communication algorithms
  - These algorithms exhibit the property of "fixed" computation on a large data set.
- Glueless MP support
  - Up to 4 CPUS can co-exist without external arbiters
- Large cache memories
- Faster external busses

Pentium III

- 1999, 9.5M transistors, 0.25μ, 450 MHz… 1.5GHz
- SSE instructions
  - Streaming SIMD Extensions
  - Allows operations on four single-precision floating-point numbers in one instruction
  - Applications related to 3D graphics, benefit
- Increased L2 cache (256KB and 512KB)
- External bus at 100 /133 MHz

Pentium-4

- 2000, 18M transistors, 0.18μ, 1.4… 3.06 GHz
- Supposed to be last of the x86 based implementation
- Very deep pipelining (20 stages) allows high frequencies to be attained
- New MMX extensions called SSE2
  - 128-bit SIMD double-precision floating point and integer instructions
- External bus at 100 /133/533 MHz

NetBurst Microarchitecture

- Also called P68 , was the successor to the P6 microarchitecture in the x86 family of CPUs made by Intel.
- NetBurst is sometimes referred to as the Intel P7, Intel 80786
- 20 stage pipeline is a significant increase in the number of stages when compared to the Pentium 3 which had only 10 stages in its pipeline.
- The Prescott architecture, the last core of the Pentium 4, has a 31 stage pipeline.

NetBurst Microarchitecture

- A drawback of having more stages in a pipeline is an increase in the number of stages that need to be traced back in the event that the branch predictor makes a mistake, increasing the penalty paid for a mis-prediction.
- To address this issue
  - Intel devised the "Rapid Execution Engine“
    - the ALUs in the core of the CPU actually operate at twice the core clock frequency.
  - Improved branch prediction technology

NetBurst Microarchitecture

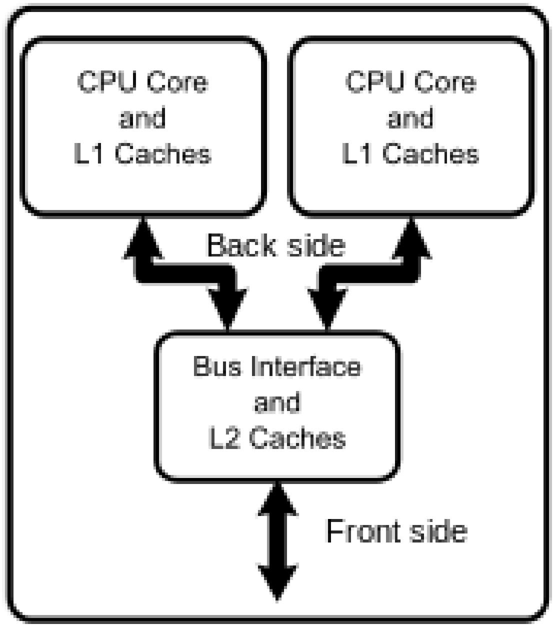
- With this architecture, Intel was looking to touch speeds of 10 GHz,
- Rising clock speed resulted in increasing problems with keeping power dissipation within acceptable limits.
- Intel reached limits at a speed of 3.8 GHz
- As a result, Intel decided to abandon NetBurst, and has since developed a newer microarchitecture, known as Core microarchitecture

Core microarchitecture

- Multi-core processor microarchitecture unveiled by Intel in 2006 It is based around Intel P6 microarchitecture, which traces its history back to the 1995 Pentium Pro.
- The architecture features lower power usage
- It supports Intel 64 (Intel's implementation of x86-64) and SSSE3.
- The pipeline is 14 stages long
- Core's execution unit is 4-issues wide, compared to the 3-issue cores of P6 and NetBurst microarchitectures.
- Multicore core design with linked L1 cache and shared L2 cache engineered for maximum performance per watt and improved scalability.

Multicore Processor

- Single computing component with two or more independent processing units
- Each unit is called cores, which read and execute program instructions



Source: Wikipedia.

Core 2

- A brand encompassing a range of Intel's consumer 64-bit x86-64
  - single-, dual-, and quad-core CPUs
- The single- and dual-core models are single-die, whereas the quad-core models comprise two dies, each containing two cores, packaged in a multi-chip module
- The Core microarchitecture returned to lower clock rates and improved the usage of both available clock cycles and power when compared with the preceding NetBurst microarchitecture of the Pentium 4/D-branded CPUs.

Core 2

- The Core microarchitecture provides more efficient decoding stages, execution units, caches, and buses, reducing the power consumption of Core 2-branded CPUs while increasing their processing capacity.

Core i5 (Nehelam)

- 45 nm process technology
- 4 physical cores
- 32+32 Kb (per core) L1 cache
- 256 Kb (per core) L2 cache
- 8 MB common L3 cache
- Introduced September 8, 2009
- 2-channels DDR3

Skylake

- Introduced August 2015
- 14 nm feature size
- L1 cache 64 KB per core
- L2 cache 256 KB per core
- L3 cache 8192 KB shared
- Up to four cores as the default mainstream configuration
- Support for both DDR3L SDRAM and DDR4 SDRAM

Skylake

- Maximum Thermal design power (TDP) up to 95 W
- Support for 16 PCI Express 3.0 lanes from CPU,
- Support for Thunderbolt 3.0
- Intel MPX (Memory Protection Extensions)
- Integrated Gen9 GPU supports Direct3D V 12
  - Full fixed function High Efficiency Video Coding (HEVC) encoding/decoding acceleration.
  - JPEG encoding acceleration for resolutions up to 16,000x16,000 pixels.

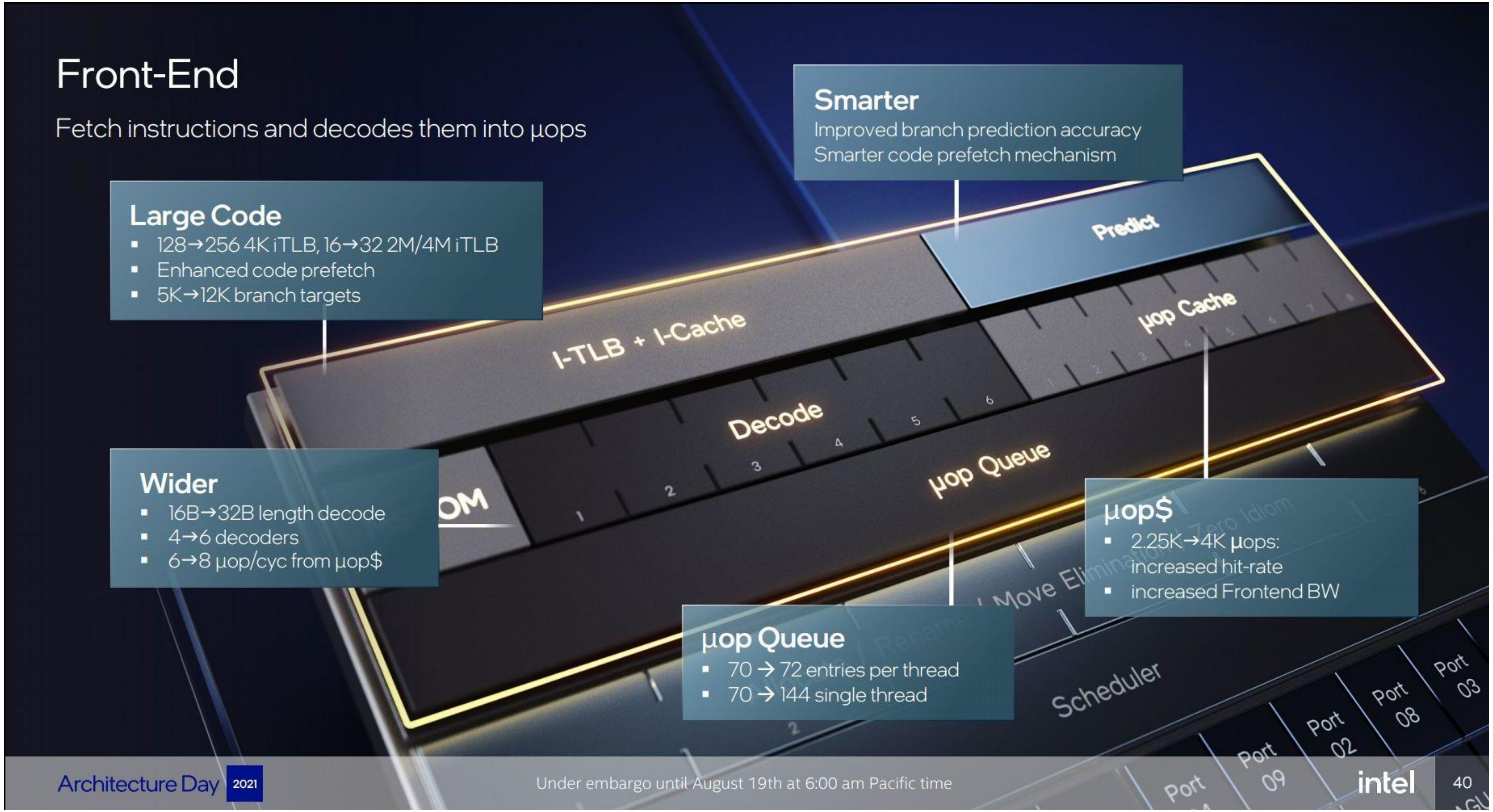
Sapphire Rapid

- Sapphire Rapids is a codename for Intel's server (fourth generation Xeon Scalable) and workstation processors based on Intel 7nM technology
- Up to 60 Golden Cove CPU cores per package
- AVX512-FP16[21]
- Advanced Matrix Extensions (AMX)
- Trust Domain Extensions (TDX), a collection of technologies to help deploy hardware-isolated virtual machines (VMs) called trust domains (TDs)

Sapphire Rapid

- L1 cache 112 KB per core (64 KB instruction+ 48 KB data)
- L2 cache 2 MB per core
- L3 cache Up to 112.5 MB
- L4 cache Up to 64 GB
- Golden Cove
  - Golden Cove is a codename for a CPU microarchitecture developed by Intel and released in November 2021
  - It is fabricated using Intel's Intel 7 process node, previously referred to as 10 nm Enhanced SuperFin (10ESF)

Golden Cove Microarchitecture



Front-End

Fetch instructions and decodes them into μops

Large Code
- 128→256 4K iTLB, 16→32 2M/4M iTLB
- Enhanced code prefetch
- 5K→12K branch targets

Smarter

Improved branch prediction accuracy
Smarter code prefetch mechanism

Wider
- 16B→32B length decode
- 4→6 decoders
- 6→8 μop/cyc from μop$

μop$
- 2.25K→4K μops: increased hit-rate
- increased Frontend BW

μop Queue
- 70 → 72 entries per thread
- 70 → 144 single thread

Architecture Day 2021

Under embargo until August 19th at 6:00 am Pacific time

intel 40

Intel 64 bit Desktop Processors

- 1st generation – Nehalem microarchitecture
- 2nd generation - Sandy Bridge microarchitecture
- 3rd generation - Ivy Bridge microarchitecture
- 4th generation - Haswell microarchitecture
- 5th generation – Broadwell microarchitecture
- 6th generation – Skylake microarchitecture
- 7th generation – Kaby Lake microarchitecture
- 8th generation - Coffee Lake microarchitecture
- 9th generation - Cannon Lake microarchitecture
- 10th generation – Ice Lake microarchitecture
- 11th generation – Tiger Lake microarchitecture
- 12th generation – Alder Lake microarchitecture

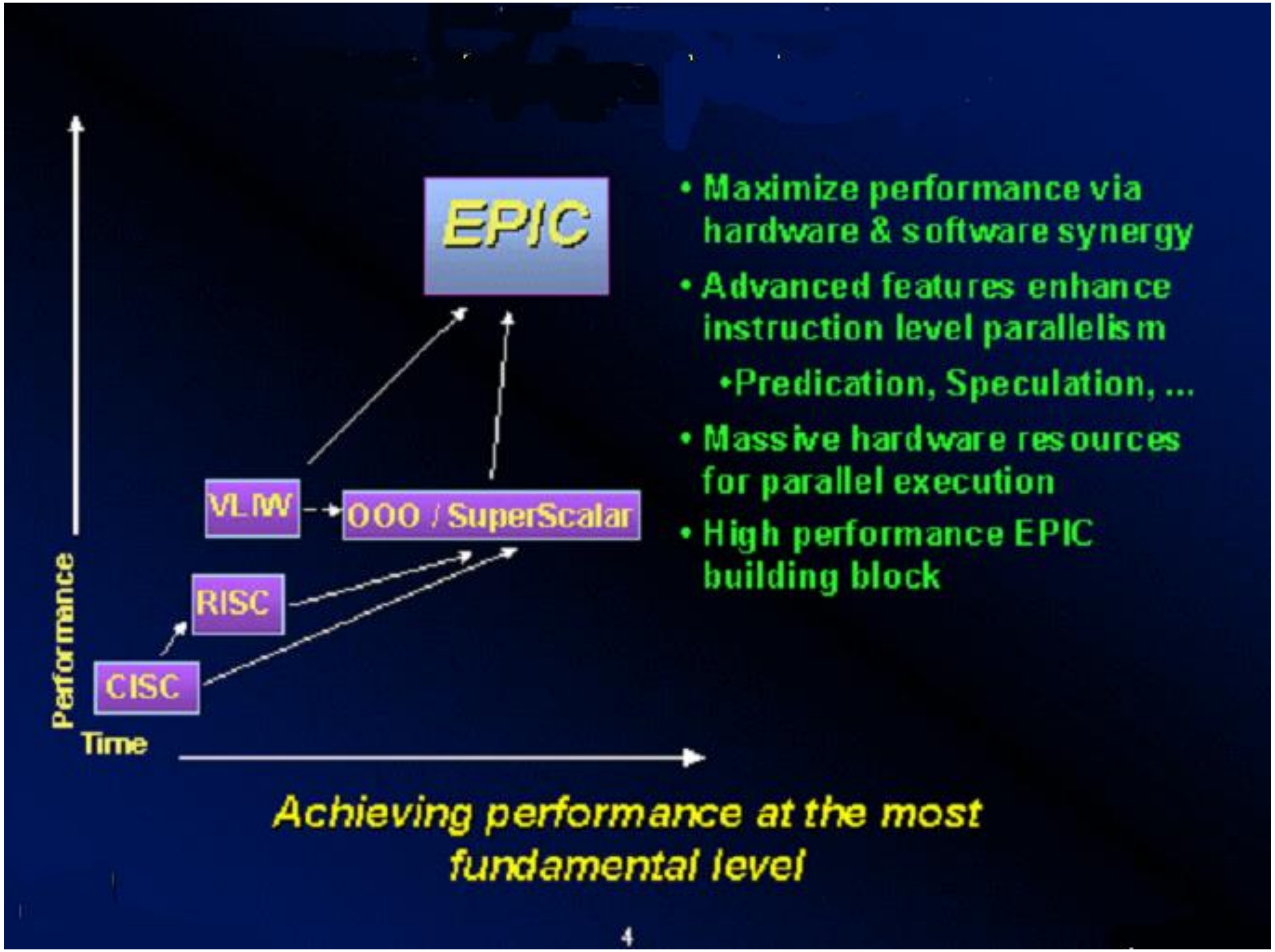
AMD64

- AMD64 was created as an alternative to Intel and Hewlett Packard IA-64 architecture.
- Originally announced in 1999, the architecture was positioned by AMD as a new way to add 64-bit computing capabilities to the existing x86 architecture
    - as opposed to Intel's approach of creating an entirely new 64-bit architecture with IA-64.
- The first AMD64-based processor, the Opteron, was released in April 2003.
- The primary defining characteristic of AMD64 is the availability of 64-bit general-purpose processor registers and 64-bit virtual addresses.

Itanium (Merced) and EPIC approach

- 2001, 25.1 Million transistors, 0.18µ, 733,800 MHz
- 32KB L1, 96KB L2 and 4MB L3 caches
- Intel/ HP joint effort has defined a new CPU architecture based on VLIW approach
- Does provide x86 compatibility (thru dedicated hardware)
- Instruction format redefined into EPIC (Explicitly Parallel Instruction Computing)
    - Instruction encodings tell the CPU which parts of the chip will be used to process data
- McKinley, Madison and Deerfield to follow

EPIC Design Philosophy



- Maximize performance via hardware & software synergy
- Advanced features enhance instruction level parallelism
    - Predication, Speculation, ...
- Massive hardware resources for parallel execution
- High performance EPIC building block

Achieving performance at the most fundamental level

4

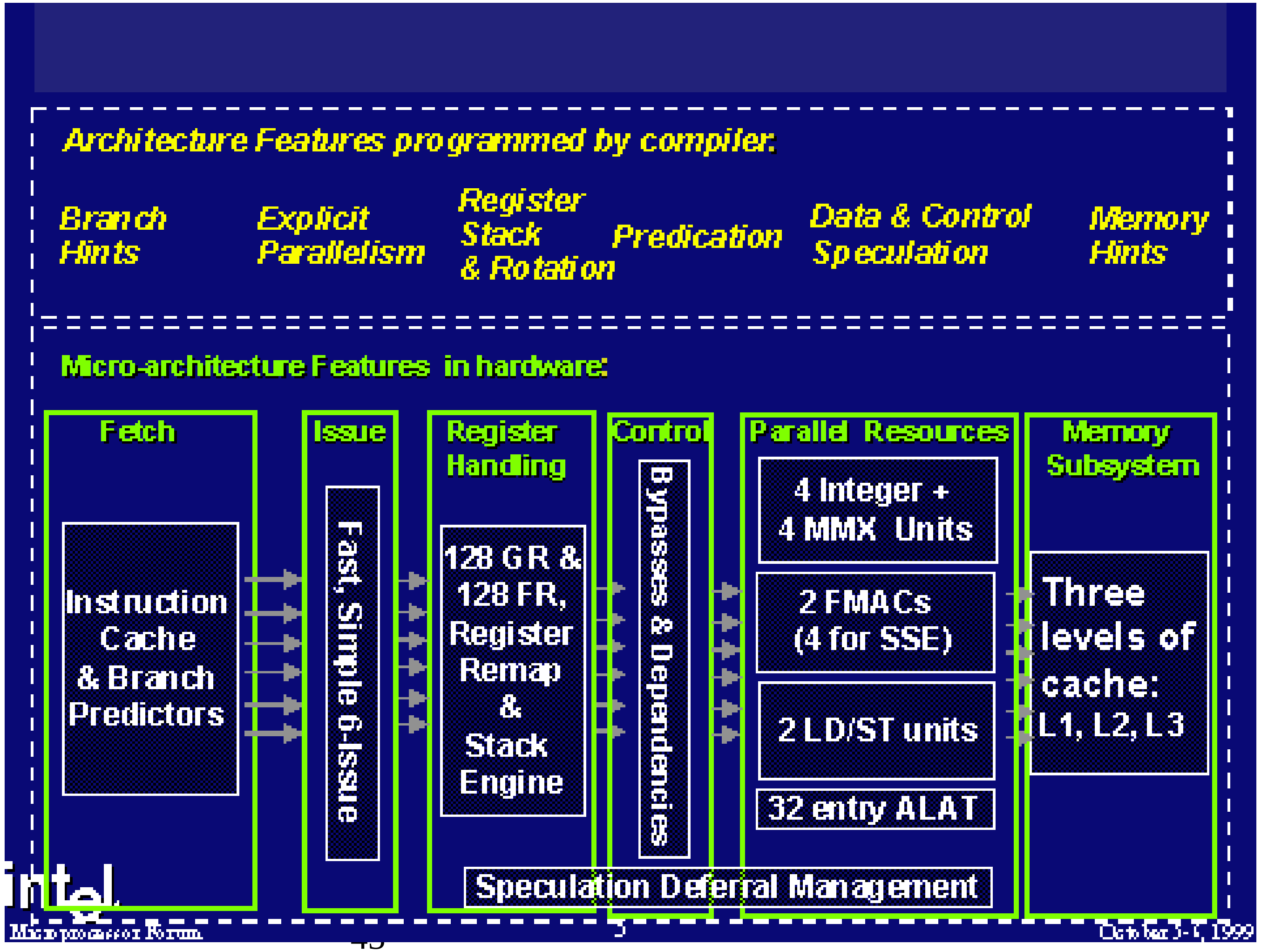
VLIW Processors

- Considerable number of EUs say 5-30
- Control word length of 16-32 bit / unit
- Instruction word length of 100 to 1024 bits
- Instructions are scheduled statically by the compiler
  - Removes the burden from the processor
- Lesser complexity is exploited for
  - Raising clock speed or parallelism or both
- Disadvantage
  - Complier becomes complex
  - Internal details of the CPU have to exposed

VLIW Processors

- Disadvantage
  - Same complier may not work for later members of the family which are otherwise ISA compatible
  - Complier takes into account worst delays in cases of cache miss
  - Actual hit of miss cannot be predicted, resulting in performance hit
  - Wasted memory space and bandwidth due to bubbles in instruction word
  - Assembly language programming is almost ruled out

Itanium Architecture



Architecture Features programmed by compiler:

Branch Hints

Explicit Parallelism

Register Stack & Rotation

Predication

Data & Control Speculation

Memory Hints

Micro-architecture Features in hardware:

Fetch

Instruction Cache & Branch Predictors

Issue

Fast, Simple 6-Issue

Register Handling

128 GR & 128 FR, Register Remap & Stack Engine

Control

Bypasses & Dependencies

Parallel Resources

4 Integer + 4 MMX Units

2 FMACs (4 for SSE)

2 LD/ST units

32 entry ALAT

Memory Subsystem

Three levels of cache: L1, L2, L3

Speculation Deferral Management

intel

Microprocessor Forum

5

October 5-7, 1999

Thank You

ashishk @ cdac.in