Intel® Itanium™ Processor Microarchitecture Overview

Harsh Sharangpani

Principal Engineer and IA-64 Microarchitecture Manager

Intel Corporation

Unveiling the Intel® Itanium™ Processor Design

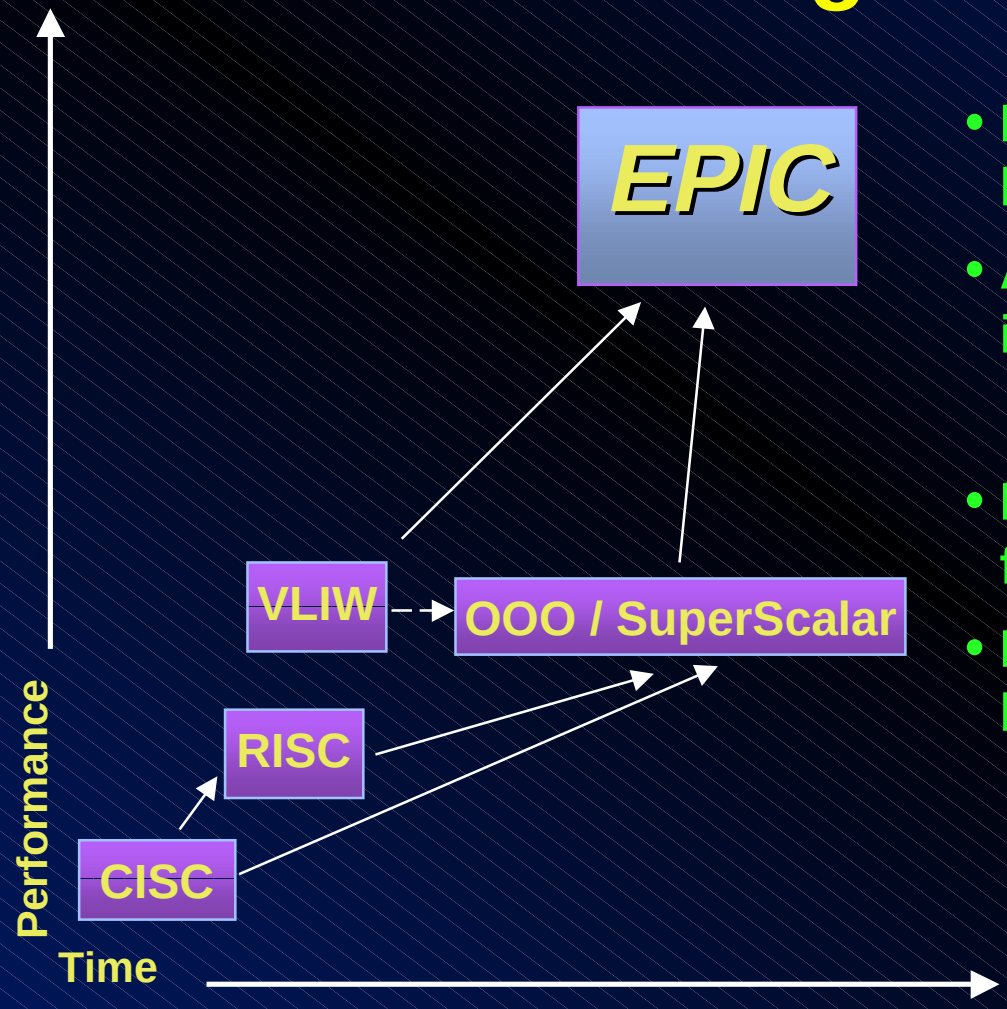
- Leading-edge implementation of IA-64 architecture for world-class performance
- New capabilities for systems that fuel the Internet Economy
- Strong progress on initial silicon

Itanium™ Processor Goals

- World-class performance on high-end applications
  - High performance for commercial servers
  - Supercomputer-level floating point for technical workstations
- Large memory management with 64-bit addressing
- Robust support for mission critical environments
  - Enhanced error correction, detection & containment
- Full IA-32 instruction set compatibility in hardware
- Deliver across a broad range of industry requirements
  - Flexible for a variety of OEM designs and operating systems

Deliver world-class performance and features for servers & workstations and emerging internet applications

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

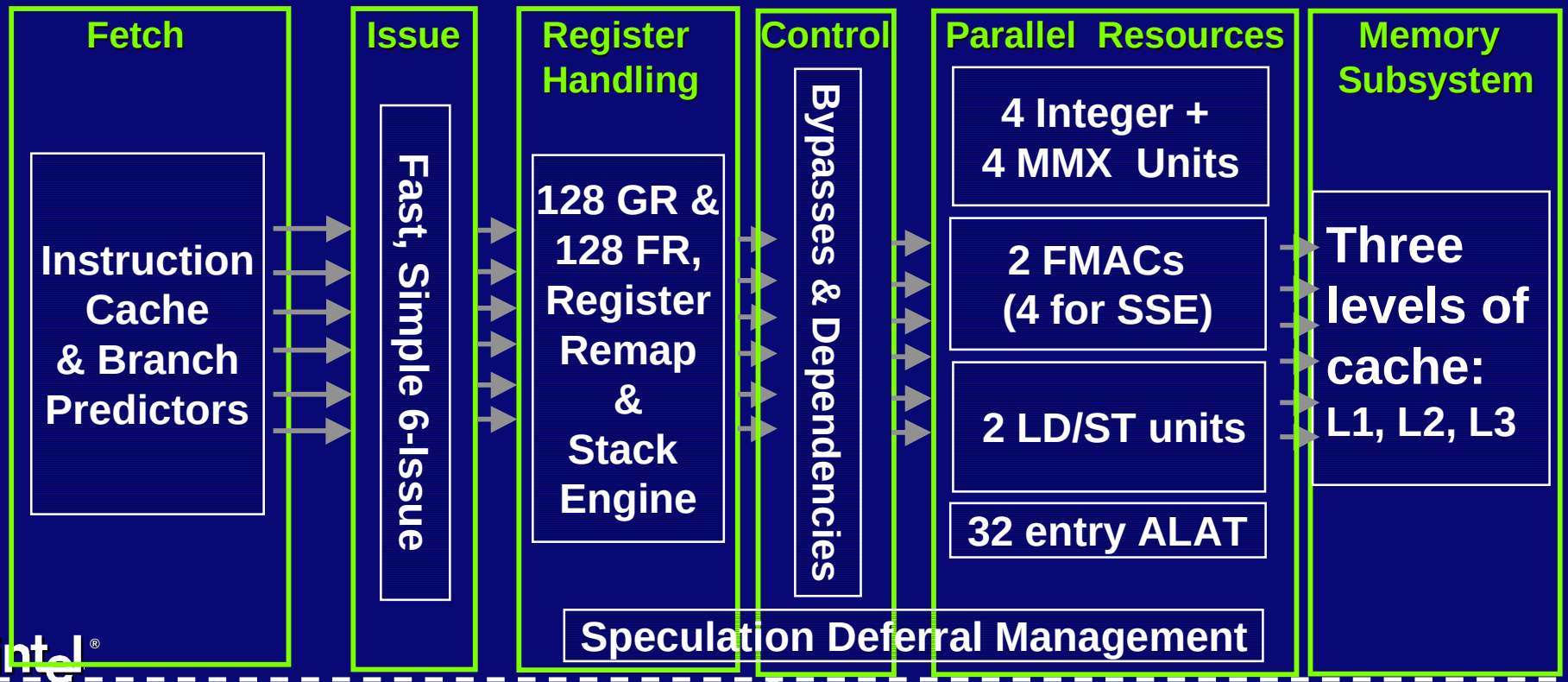
Achieving performance at the most fundamental level

Itanium™ EPIC Design Maximizes SW-HW Synergy

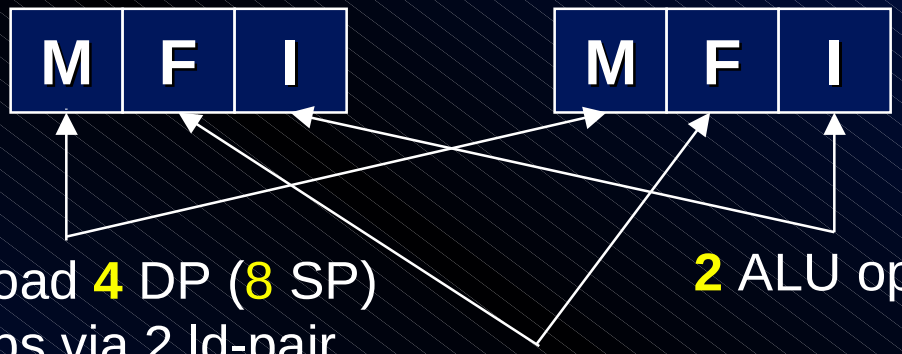
Architecture Features programmed by compiler:

Branch Hints | ***Explicit Parallelism*** | ***Register Stack & Rotation*** | ***Predication*** | ***Data & Control Speculation*** | ***Memory Hints***

Micro-architecture Features in hardware:



Breakthrough Levels of Parallelism

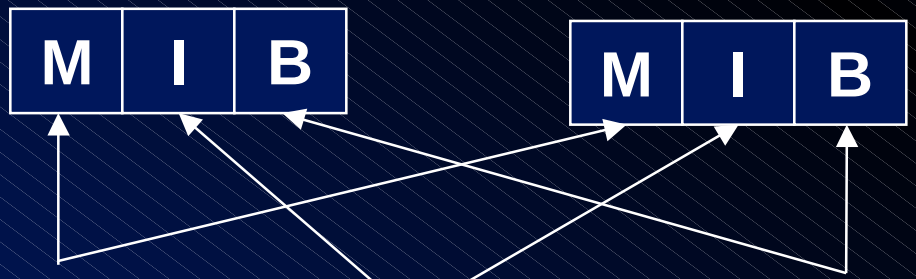


- Load **4** DP (8 SP) ops via 2 ld-pair
- **2** ALU oper (post incr)

2 ALU ops

4 DP FLOPS (**8** SP FLOPS)

6 instructions provides 12 parallel ops/clock (SP: 20 parallel ops/clock) for digital content creation & scientific computing



2 Loads + **2** ALU ops (post incr)

2 ALU ops

1 Branch Hint + **1** Branch instr

6 instructions provides 8 parallel ops / clock for enterprise & Internet applications

Itanium™ delivers greater instruction level parallelism than any contemporary processor

Highlights of the Itanium™ Pipeline

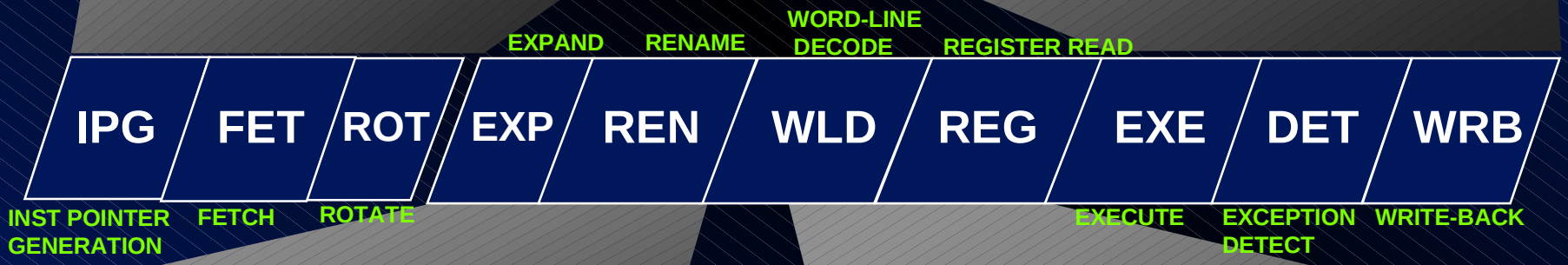
- 6-Wide EPIC hardware under precise compiler control
  - Parallel hardware and control for predication & speculation
  - Efficient mechanism for enabling register stacking & rotation
  - Software-enhanced branch prediction
- 10-stage in-order pipeline with cycle time designed for:
  - Single cycle ALU (4 ALUs globally bypassed)
  - Low latency from data cache
- Dynamic support for run-time optimization
  - Decoupled front end with prefetch to hide fetch latency
  - Aggressive branch prediction to reduce branch penalty
  - Non-blocking caches and register scoreboard to hide load latency

Parallel, deep, and dynamic pipeline designed for maximum throughput

10 Stage In-Order Core Pipeline

Front End
- Pre-fetch/Fetch of up to 6 instructions/cycle
- Hierarchy of branch predictors
- Decoupling buffer

Execution
- 4 single cycle ALUs, 2 ld/str
- Advanced load control
- Predicate delivery & branch
- Nat/Exception//Retirement

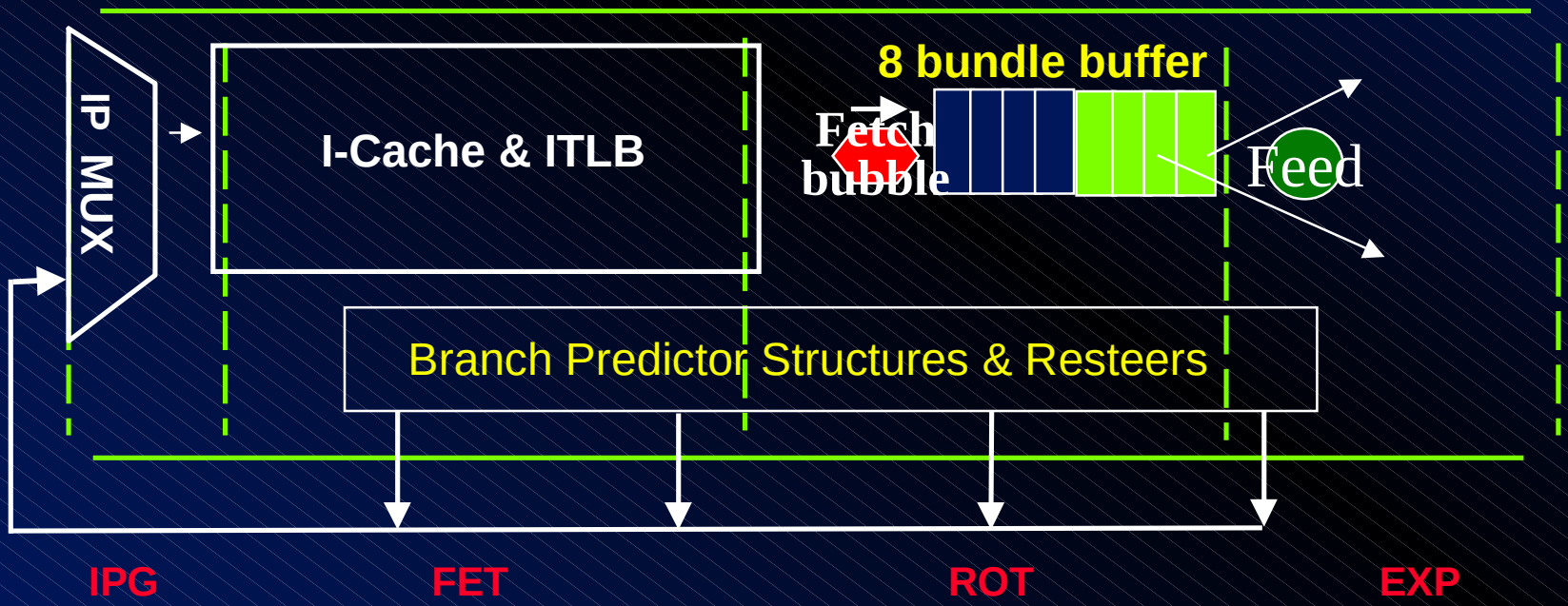


Instruction Delivery
- Dispersal of up to 6 instructions on 9 ports
- Reg. remapping
- Reg. stack engine

Operand Delivery
- Reg read + Bypasses
- Register scoreboard
- Predicated dependencies

Frontend: Prefetch & Fetch

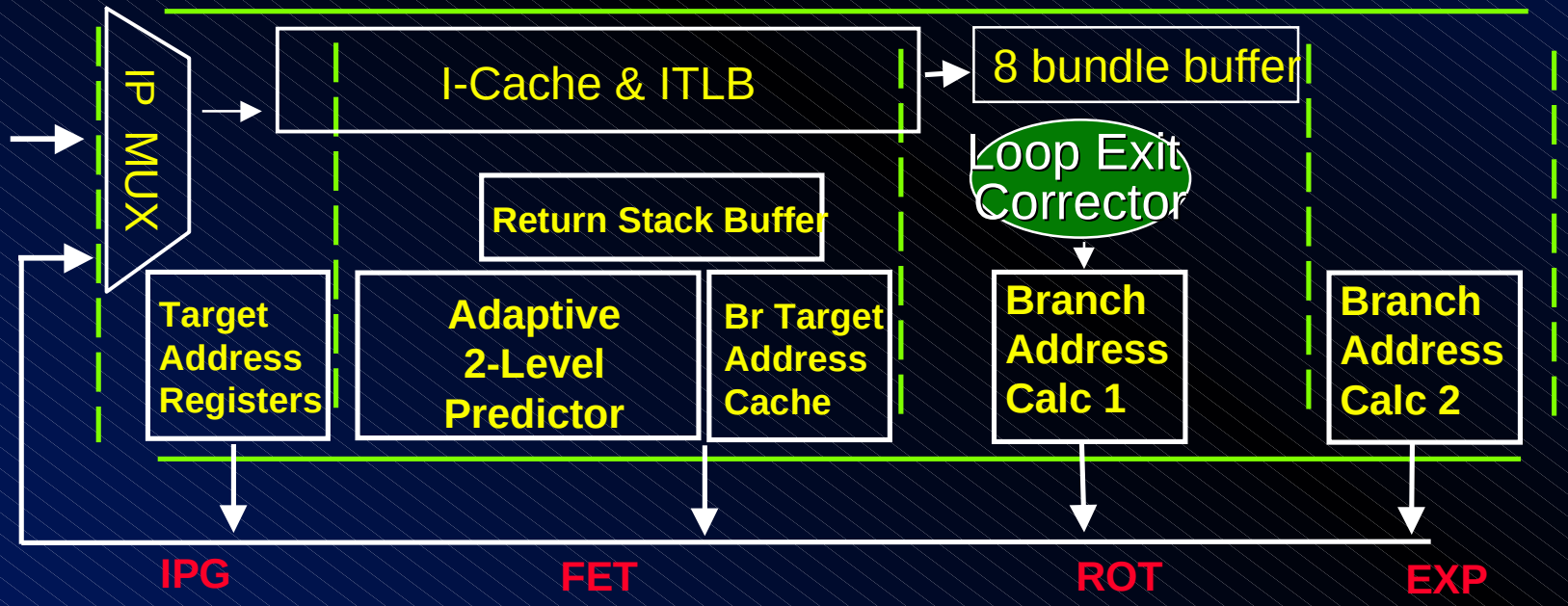
- SW-triggered prefetch loads target code early using br hints
  - Streaming prefetch of large blocks via hint on branch
  - Early prefetch of small blocks via BRP instruction
- I-Fetch of 32 Bytes/clock feeds an 8-bundle decoupling buffer
  - Buffer allows front-end to fetch even when back-end is stalled
  - Hides instruction cache misses and branch bubbles



Aggressive instruction fetch hardware to feed a highly parallel, high performance machine

Front End: Branch Prediction

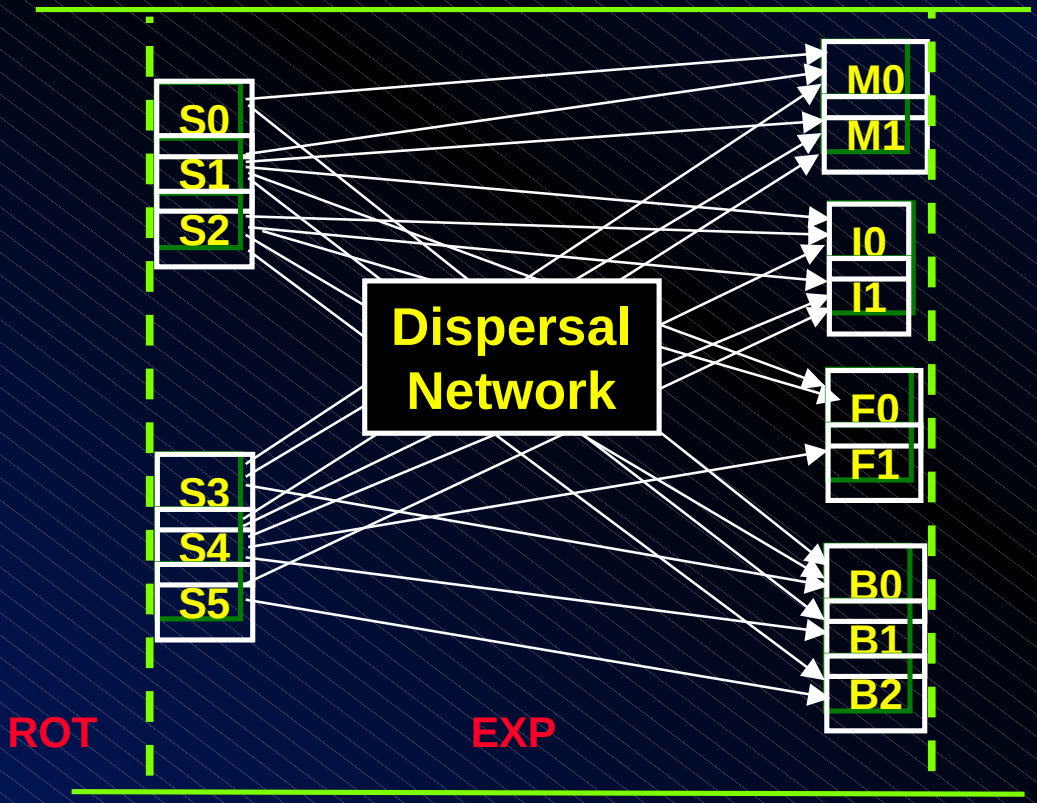
- Branch hints combine with predictor hierarchy to improve branch prediction: four progressive resteers
  - 4 TARs programmed by “importance” hints
  - 512-entry 2-level predictor provides dynamic direction prediction
  - 64-entry BTAC contains footprint of upcoming branch targets (programmed by branch hints, and allocated dynamically)



Intelligent branch prediction improves performance across all workloads

Instruction Delivery: Dispersal

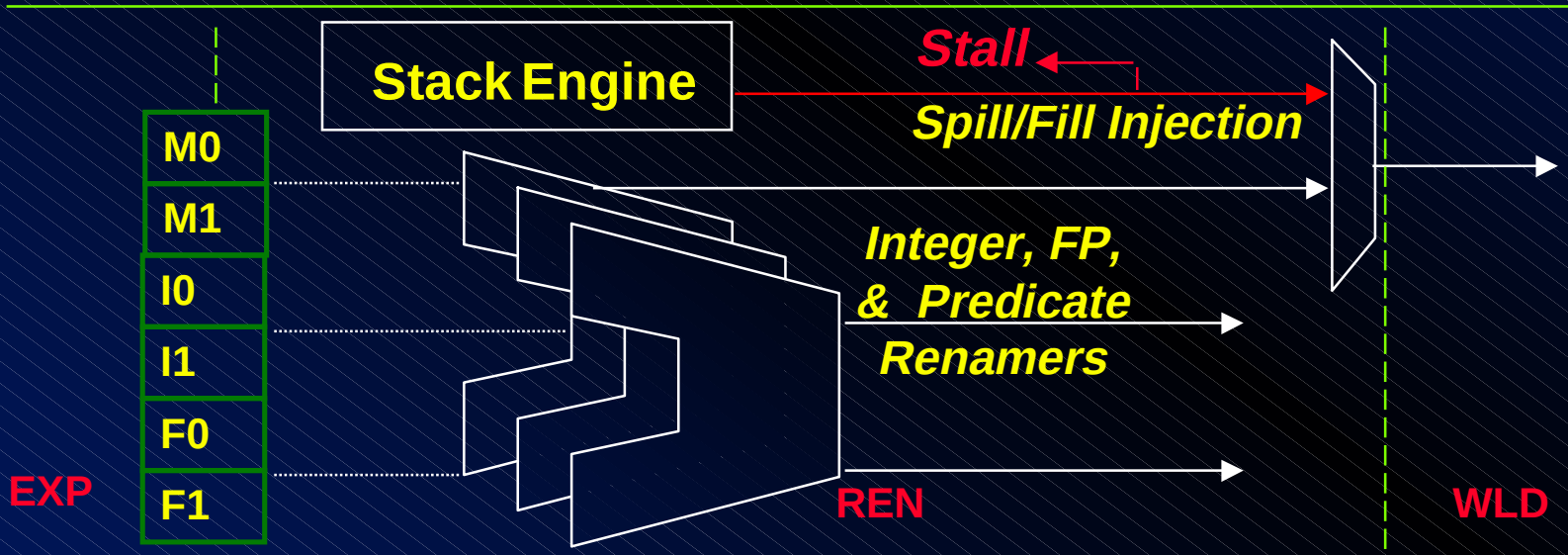
- Stop bits eliminate dependency checking
- Templates simplify routing
- 1st available dispersal from 6 syllables to 9 issue ports
  - Keep issuing until stop bit, resource oversubscription, or asymmetry



Achieves highly parallel execution with simple hardware

Instruction Delivery: Stacking

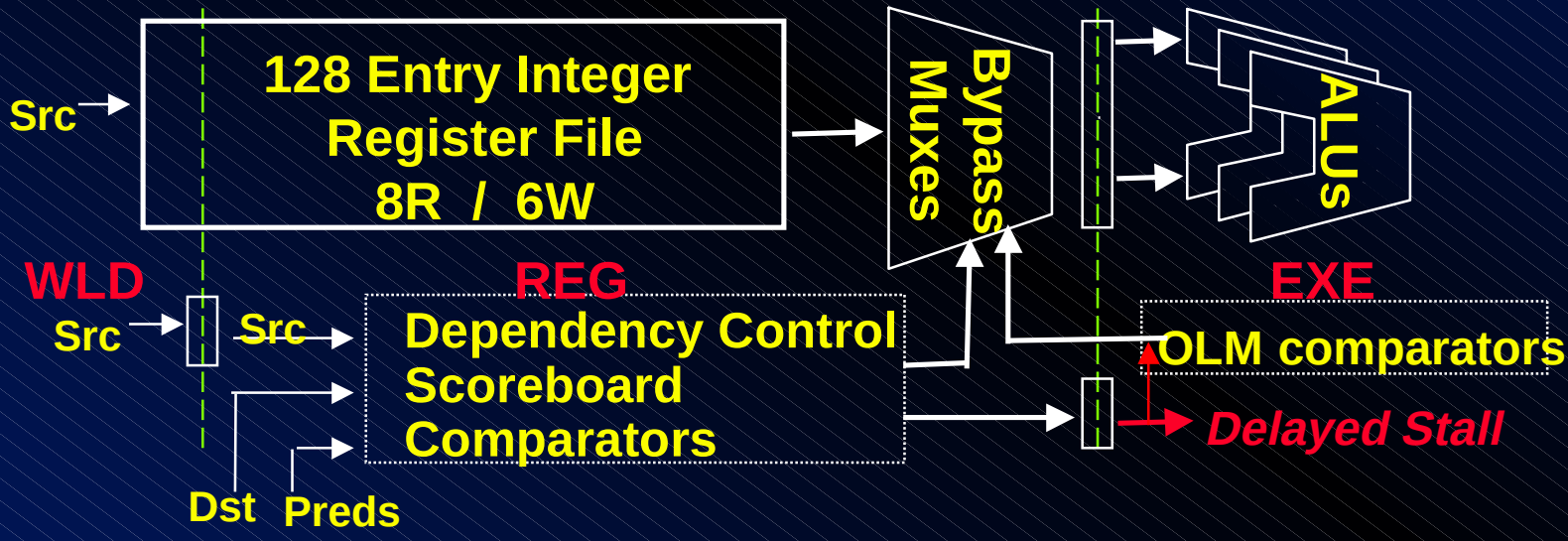
- Massive 128 register file accommodates multiple variable sized procedures via stacking
- Eliminates most register spill / fill at procedure interfaces
- Achieved transparently to the compiler
  - Using register remapping via parallel adders
  - Stack engine performs the few required spill/fills



Unique register model enables faster execution of object-oriented code

Operand Delivery

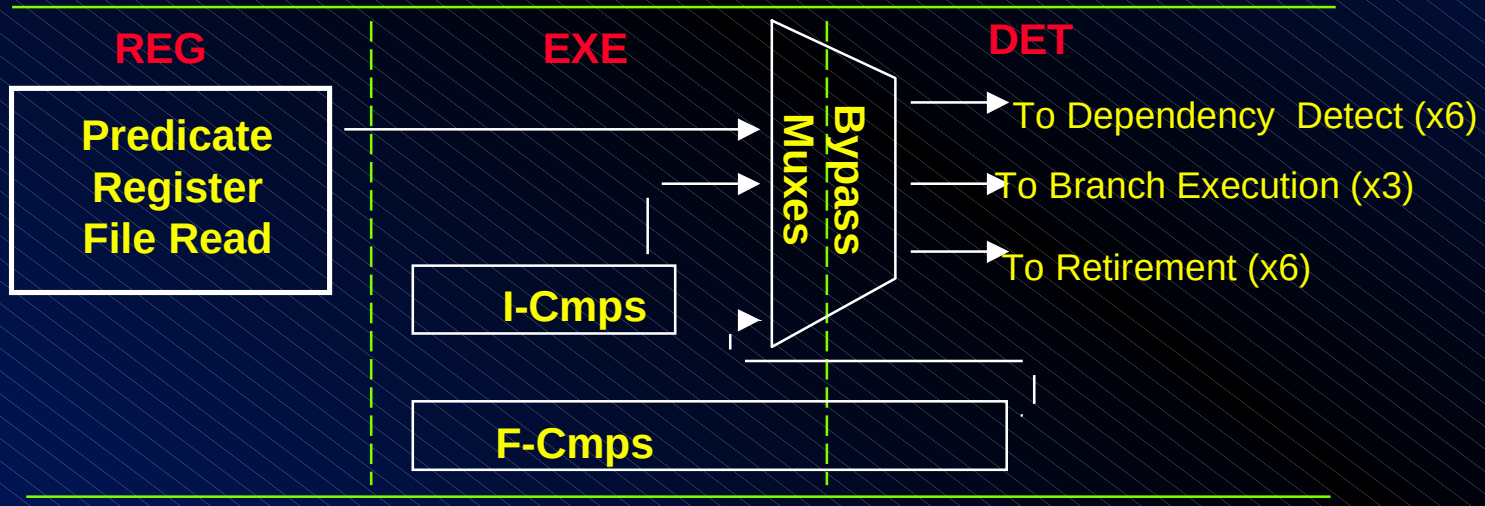
- Multiported register file + mux hierarchy delivers operands in REG
- Unique "*Delayed Stall*" mechanism used for register dependencies
  - Avoids pipeline flush or replay on unavailable data
  - Stall computed in REG, but core pipeline stalls in EXE
  - Special Operand Latch Manipulation (OLM) captures data returns into operand latches, to mimic register file read



Avoids pipeline flush to enable a more effective, higher throughput pipeline

Predicate Delivery

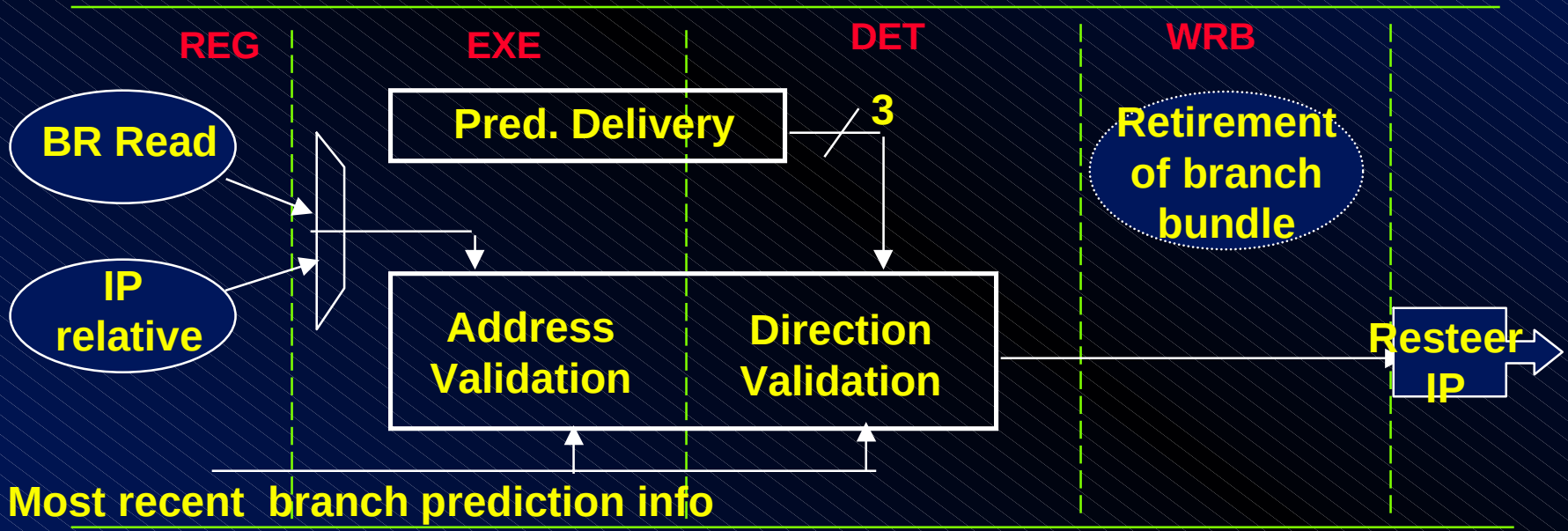
- All instructions read operands and execute
  - Canceled at retirement if predicates off
- Predicates generated in EXE (by cmps), delivered in DET, & feed into: retirement, branch execution and dependency detection
- Smart control network cancels false stalls on predicated dependencies
  - Dependency detection for cancelled producer/consumer (REG)



Higher performance through removal of branch penalties in server and workstation applications

Parallel Branch Execution

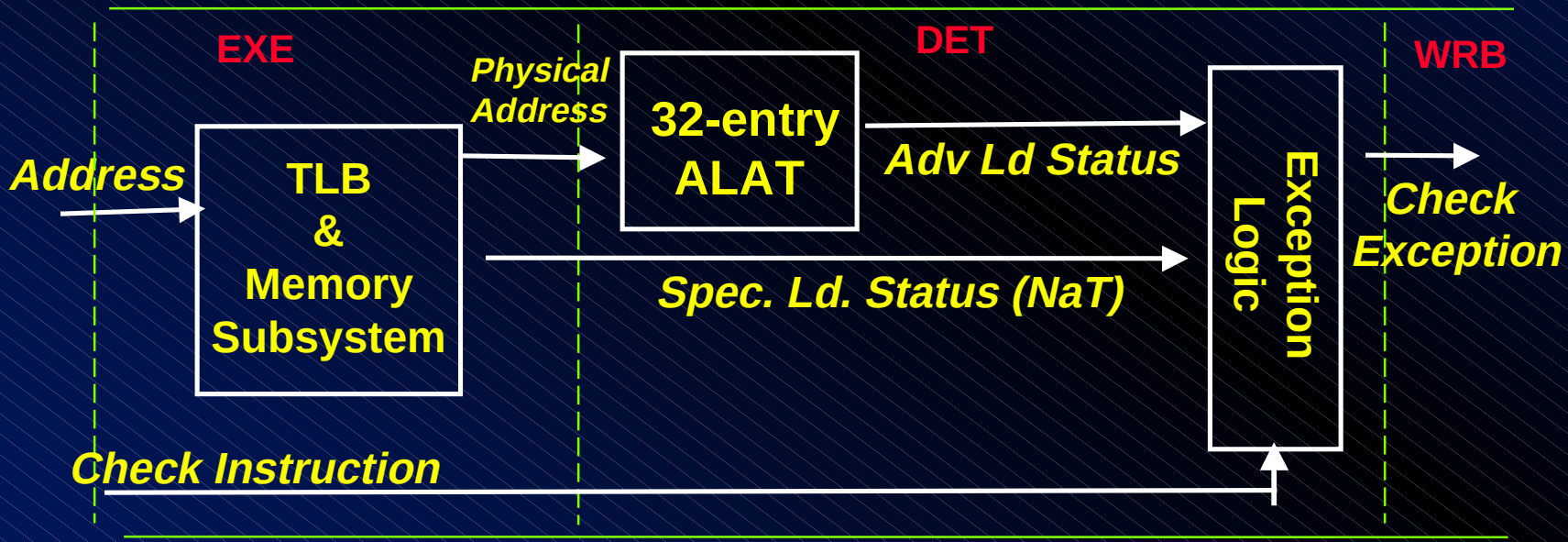
- Speculation + predication result in clusters of branches
- Execution of 3 branches/clock optimizes for clustered branches
- Branch execution in DET allows cmps-->branches in same issue group



Parallel branch hardware extends performance benefits of EPIC technology

Speculation Hardware

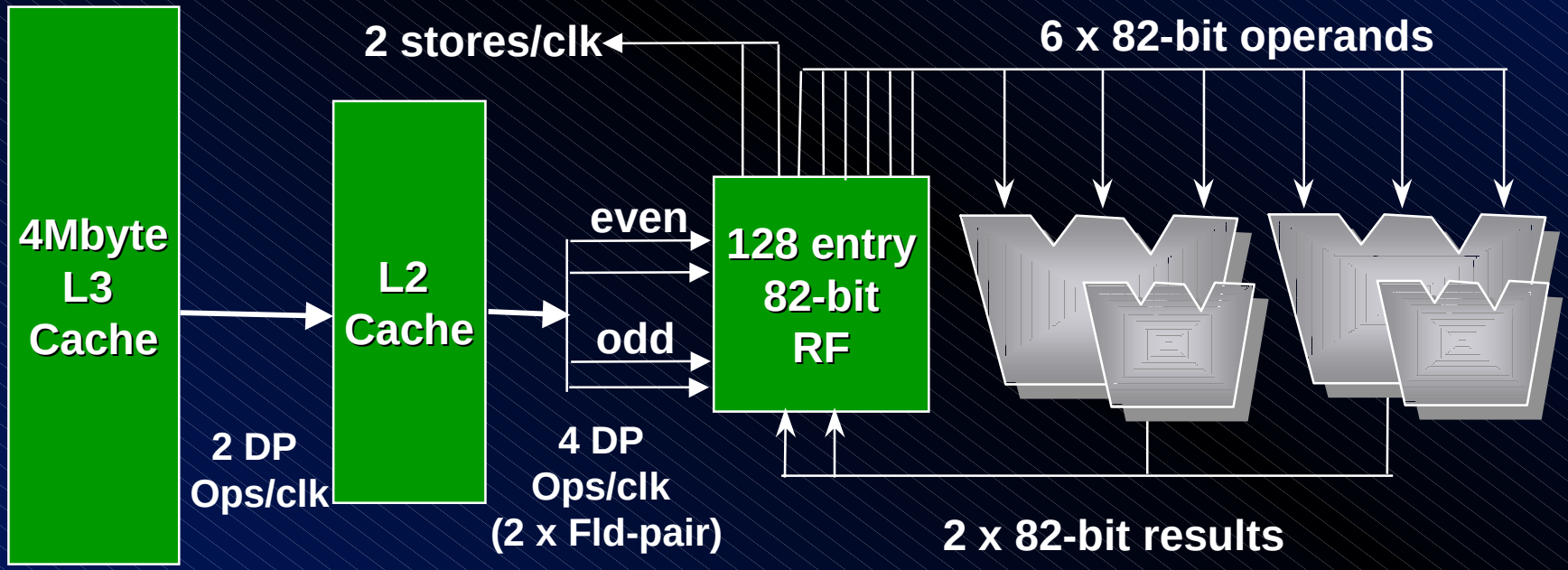
- Control Speculation support requires minimal hardware
  - Computed memory exception delivered with data as tokens (NaTs)
  - NaTs propagate through subsequent executions like source data
- Data Speculation enabled efficiently via ALAT structure
  - 32 outstanding advanced loads
  - Indexed by reg-ids, keeps partial physical address tag
- 0 clk checks: dependent use can be issued in parallel with check



Efficient elimination of memory bottlenecks

Floating Point Features

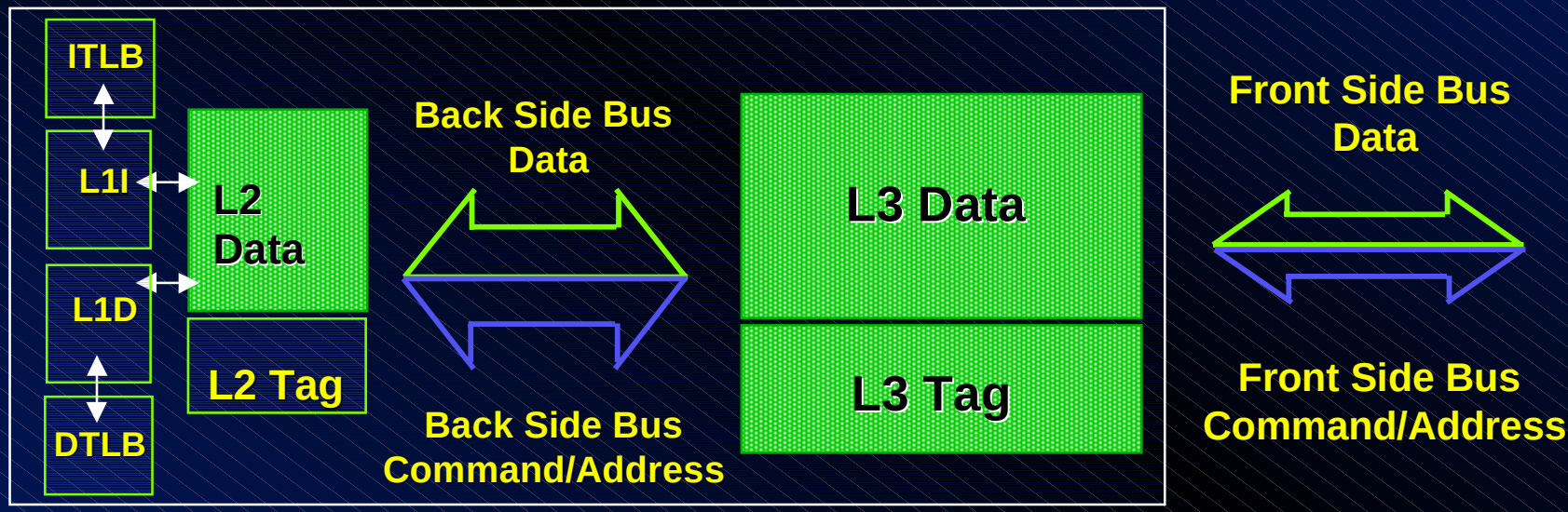
- Native 82-bit hardware provides support for multiple numeric models
- 2 Extended precision pipelined FMACs deliver 4 EP / DP FLOPs/cycle
- Performance for security and 3-D graphics
  - 2 Additional single-precision FMACs for 8 SP FLOPs/cycle (SIMD)
  - Efficient use of hardware: Integer multiply-add and s/w divide
- Balanced with plenty of operand bandwidth from registers / memory



Itanium™ processor delivers industry-leading floating point performance

Reliability & Availability Features

- Extensive Parity/ECC coverage on processor and bus
  - L3 MESI state bits sparsely encoded to protect the M-state
  - Frontside bus uses special ECC encoding for consecutive 4-bit errors



1x ECC Correction, 2 x ECC detection
Parity coverage w/ enhanced MCA

Comprehensive integrity for high-end applications

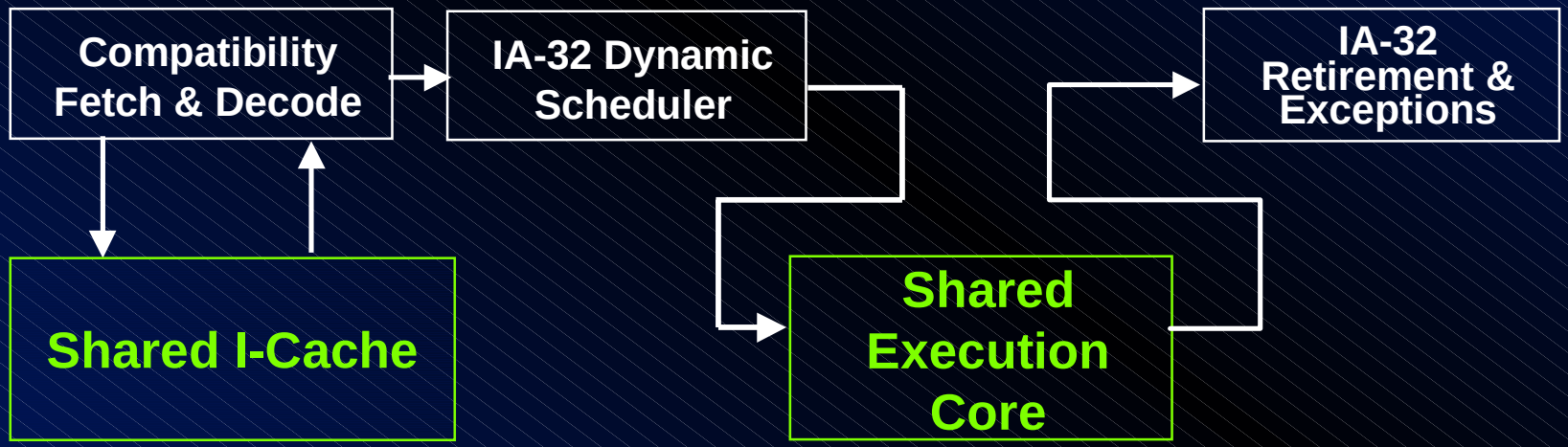
Enhanced Machine Check Architecture

	Error Type	Signaling	Example	Benefit
CONTINUE	Corrected by CPU; current process continues	CMCI	1xECC L2 data	Enhanced Reliability & Availability
	Corrected by firmware; current process continues	CMCI	I-cache parity	Enhanced Reliability & Availability
RECOVER	Affected process terminated by f/w to OS; OS is stable	LMCA	Poisoned data	Enhanced Availability
CONTAIN	Error is contained, affected node is taken off-line	GMCA	System Bus Address parity	Enhanced Reliability

Itanium™ processor delivers the mission-critical reliability required by E-business

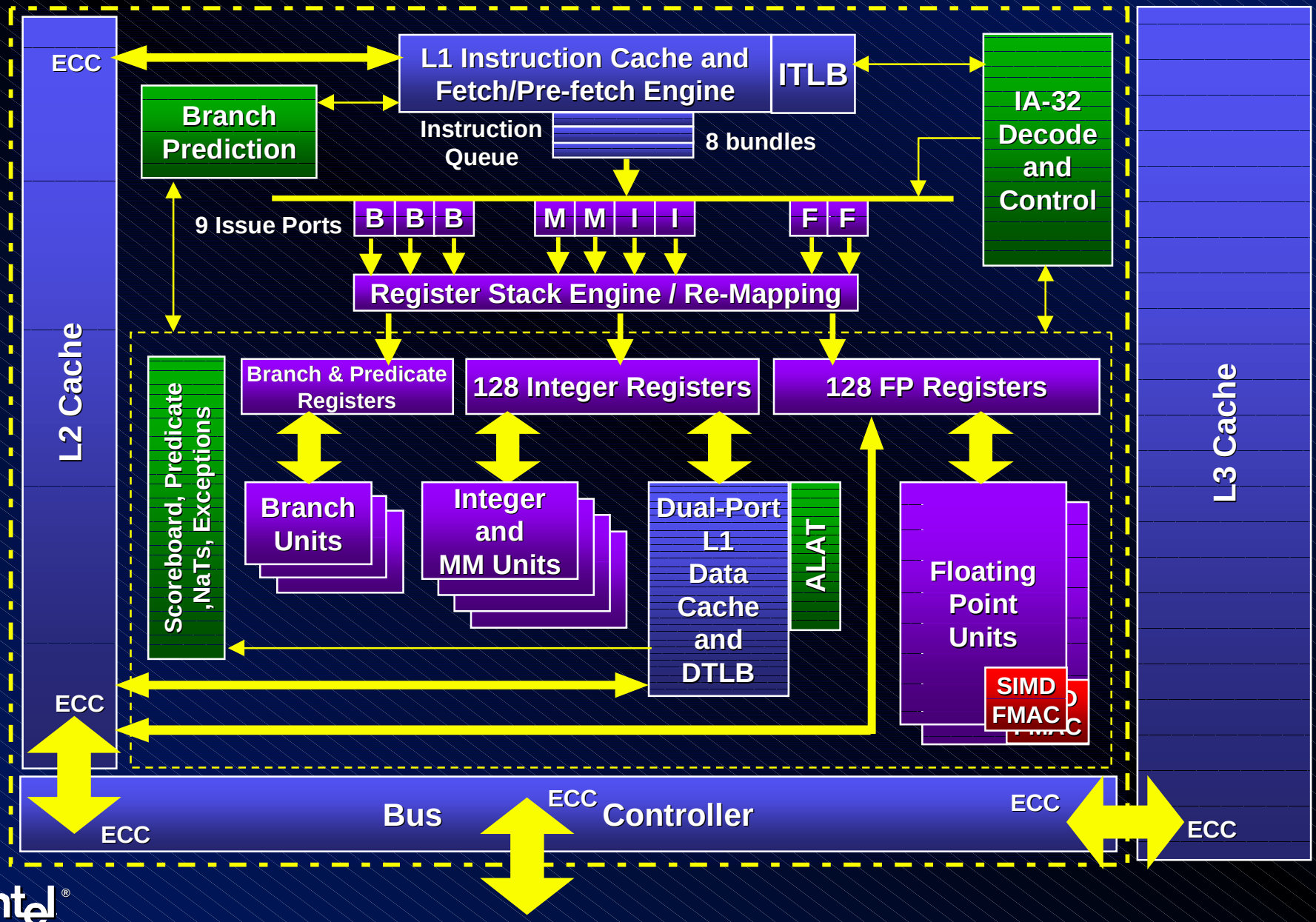
IA-32 Compatibility

- Itanium™ directly executes IA-32 binary code
  - Shared caches & execution core increases area efficiency
  - Dynamic scheduler optimizes performance on legacy binaries
- Seamless Architecture allows full Itanium performance on IA-32 system functions



Full, efficient IA-32 instruction compatibility in hardware

Intel® Itanium™ Processor Block Diagram



Itanium™ Processor Status

- **Solid progress in weeks following Itanium™ first silicon**
  - More than 4 operating systems running today
  - Demonstrated 64-bit Windows 2000 and Linux running apps
  - Initial engineering samples shipped to OEMs
- **Comprehensive functional validation underway**
  - Thorough pre-silicon functional testing included OS kernel on Itanium logic model
  - Testing including 7 OS's & many key enterprise and scientific apps
  - Multiple Intel and OEM test platform configurations (from 2 - 64 processors)
- **Planned steps to production in mid 2000**
  - Completion of functional testing phase through end of 1999
  - Performance testing/tuning accelerates in 1H'00
  - Broad prototype system deployment by Intel and OEM's early 2000

Intel® Itanium™ Processor Summary

- **High performance leading-edge design**
  - EPIC technology provides a breakthrough in hardware/software synergy
  - Predication, speculation, register stacking, & large L3 for High-End servers
  - Supercomputer-level GFLOPs performance for technical workstations
  - 64-bit memory addressability for large data sets
- **Mission-critical reliability and availability**
  - Machine check implementation maximizes error containment and correction
  - Comprehensive data integrity for e-Business, Internet and enterprise servers
- **Full IA-32 instruction level compatibility in hardware**
- **Strong Itanium™ silicon progress and industry support**