



**COMPAQ**

# ALPHA 21264 Introduction

## I- Stream

Dharmesh Parikh

# System Overview

- RISC instruction Set
- 64-bit processor
- 15 million transistors
- 1<sup>st</sup> Alpha w/ out-of-order execution
- Speculative execution

**COMPAQ**

# Alpha 21264 Key Features

- Four-wide Instruction Fetch
- Line and Branch Predictor
  - Tournament prediction using both *local* & *global* history
  - Dynamic JSR/JMP prediction
- Out-of-Order Execution Pipelines
  - Quad-speculative-issue integer pipeline
  - Dual-speculative-issue floating-point pipeline
    - ◆ One ADD, One MULTIPLY
    - ◆ Also DIV (6 bits/cycle) and SQRT (2 bits/cycle)
- Memory System
  - Two out-of-order memory references per cycle
  - Up to 16 outstanding off-chip memory references
    - ◆ 8 fills and 8 victims

**COMPAQ**

## Alpha 21264 Key Features (2)

- 80 In-flight Instructions
- Registers: 80 Integer, 72 Floating Point
- Queue Entries: 20 Integer, 15 Floating Point
- 128-Entry Fully Associative DTB
- 128-Entry Fully Associative ITB
- 64 KB L1 On-Chip Instruction Cache
  - Virtual Index/Tag
  - 2-Way Set Predict
- 64 KB L1 On-Chip Data Cache
  - Virtual Index/Physical tag
  - 2-Way Set Associative
  - Cache-hit prediction logic enhances speculative issue

# Differences from 21164

- Out of order issue
- Smaller pipeline
- Increased memory bandwidth
- Memory references can be accessed in parallel to caches
- One pipeline for both floating point and integer operations
- 4x the bandwidth

# Previous instruction types

- Branch
- Floating point
- Memory
- Memory/Function code
- Memory/branch
- Operate
- PALcode

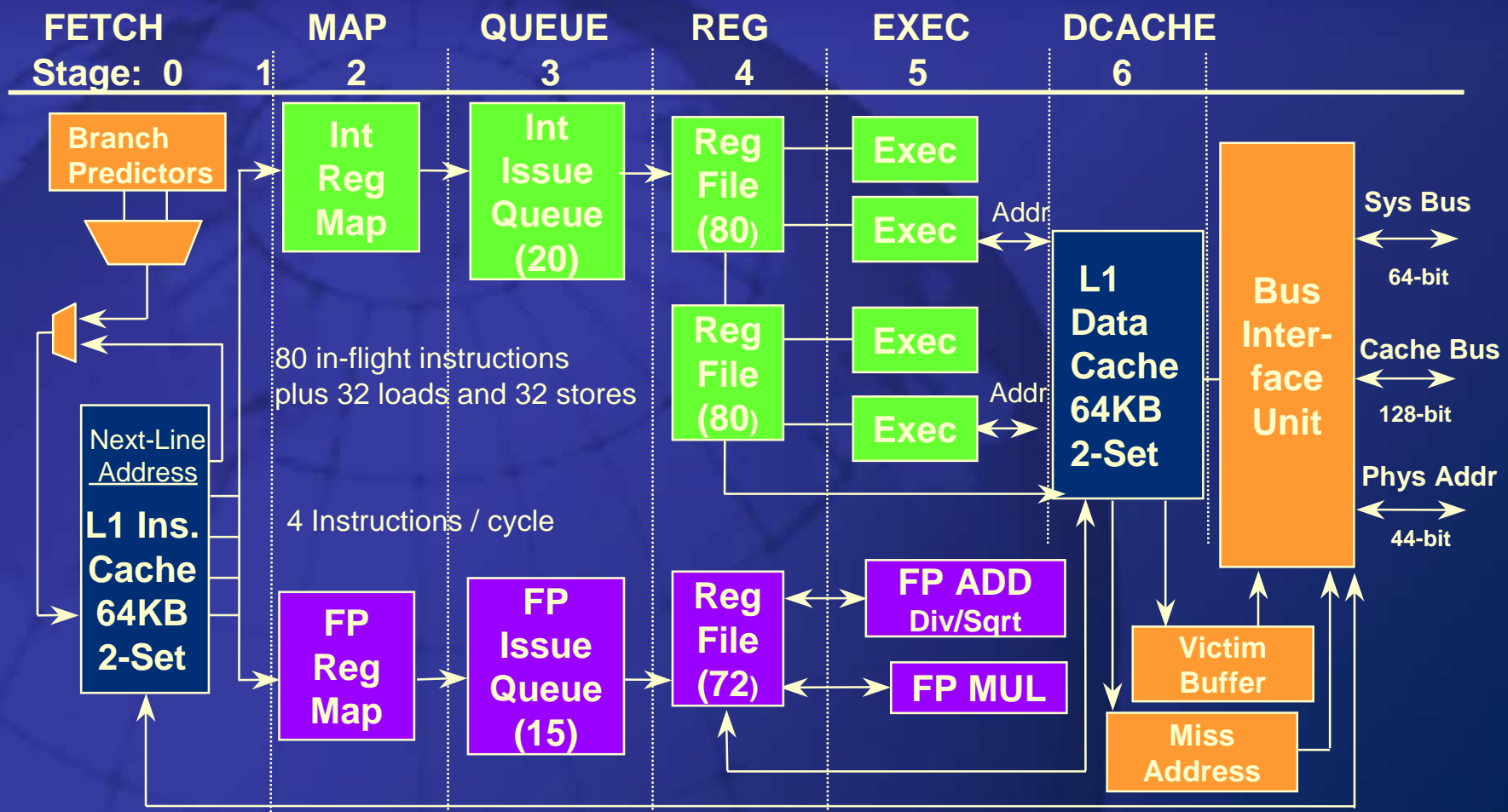
# New instructions

- Floating Point
- Cache prefetching
- MVI (motion video instructions)

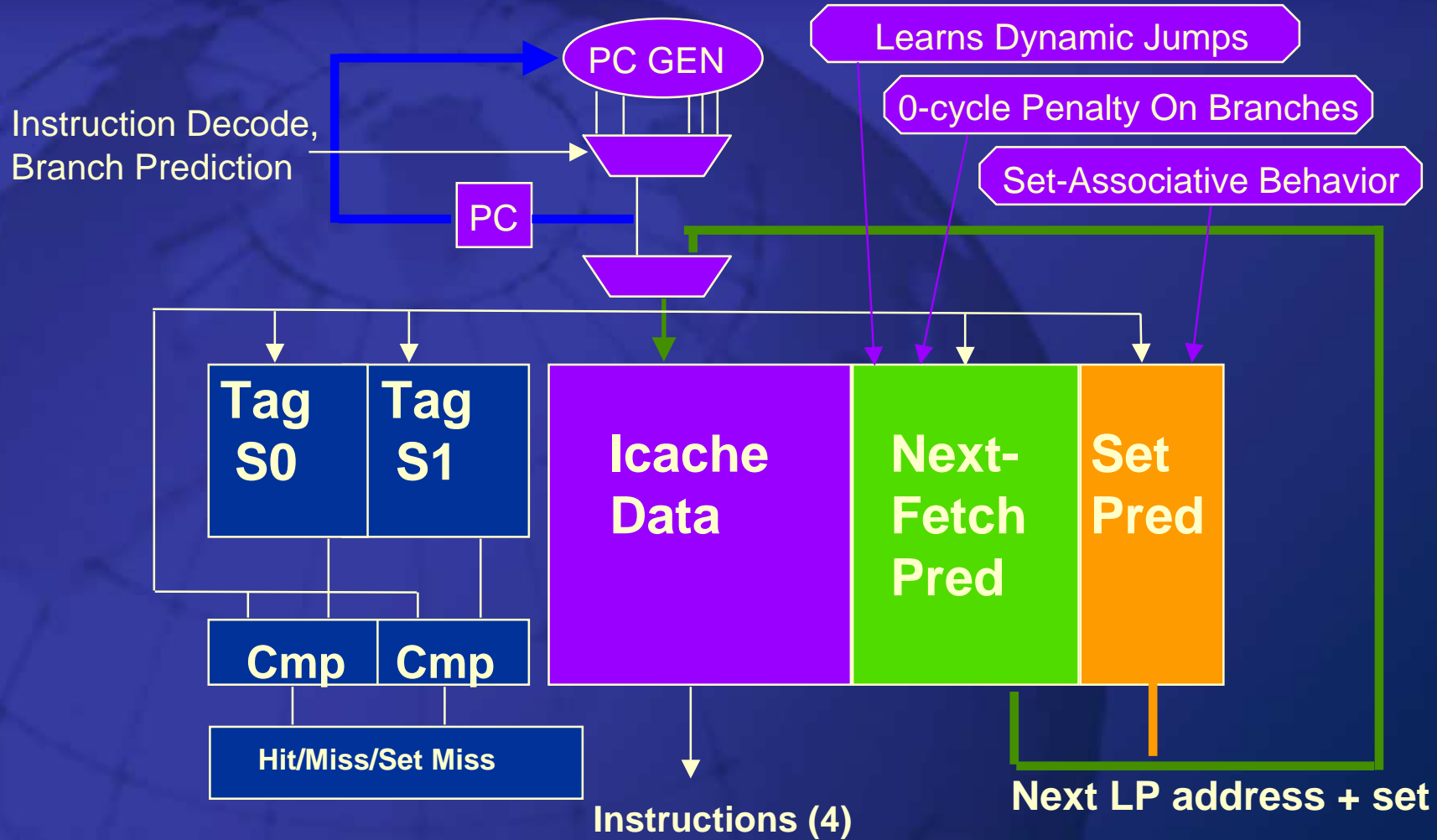
# Alpha 21264 Instruction Fetch Bandwidth Enablers

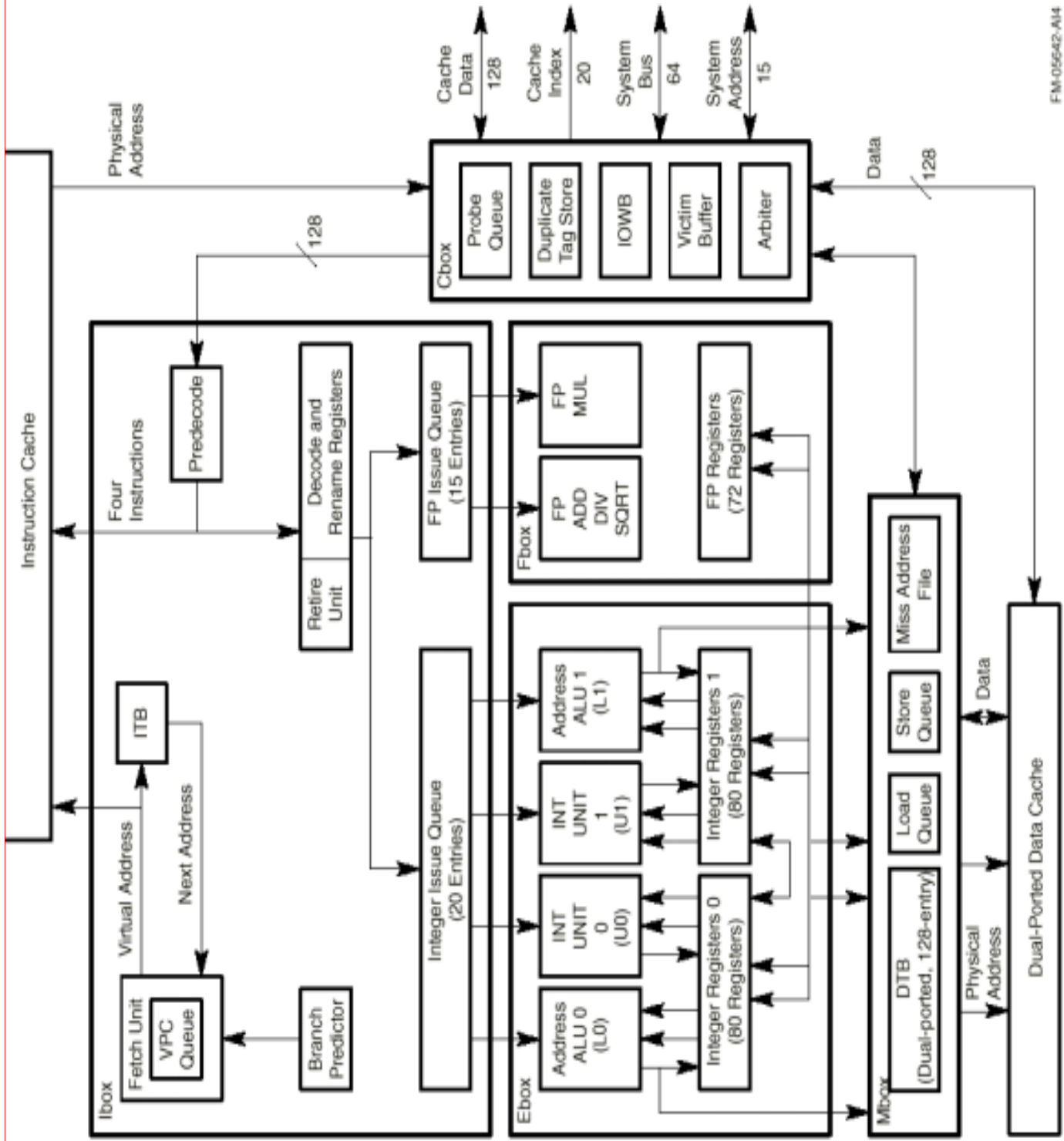
- The 64 KB two-way associative instruction cache supplies four instructions every cycle
- The *next-fetch* and *set predictors* provide fast cache access of a direct-mapped cache and eliminate bubbles in non-sequential control flows
- The instruction fetcher speculates through up to 20 branch predictions to supply a continuous stream of instructions
- The tournament branch predictor dynamically selects between *Local* and *Global* history to minimize mispredicts

# Alpha 21264 – 7 Stage Pipeline [1]



# Alpha 21264 Instruction Stream Improvements





FM-05642-A14

**COMPAQ**

3.4.0 Dual-Ported Data Cache

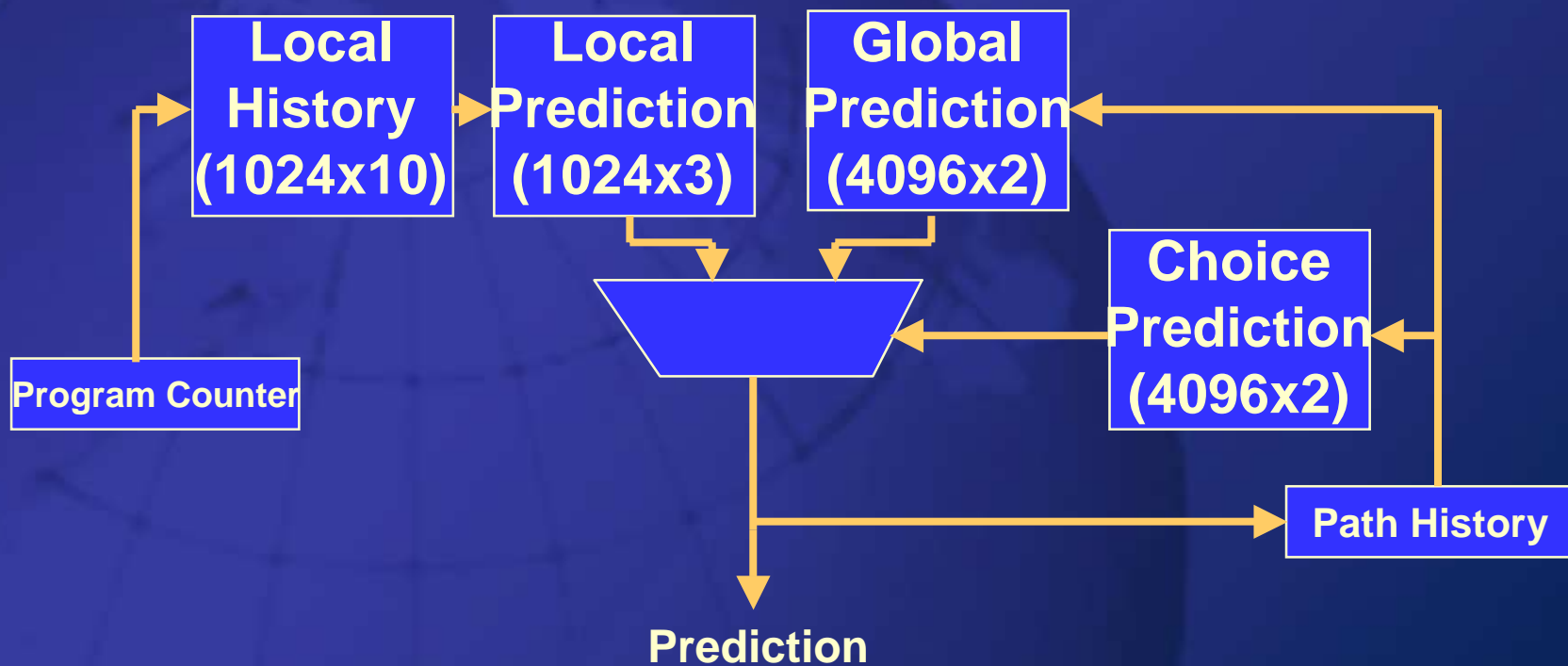
# Instruction Fetch, Issue and Retire Unit

- Virtual Program Counter Logic
- Branch Predictor
- Instruction -Stream Translation Buffer(ITB)
- Instruction fetch logic
- Register rename maps
- Integer and Floating-Point issue Queues
- Exception and Interrupt Logic
- Retire Logic

# Virtual Program Counter Logic

- The virtual program counter (VPC) logic maintains the virtual addresses for instructions that are in flight. There can be up to 80 instructions, in 20 successive fetch slots, in-flight between the register rename mappers and the end of the pipeline. The VPC logic contains a 20-entry table to store these fetched VPC addresses.

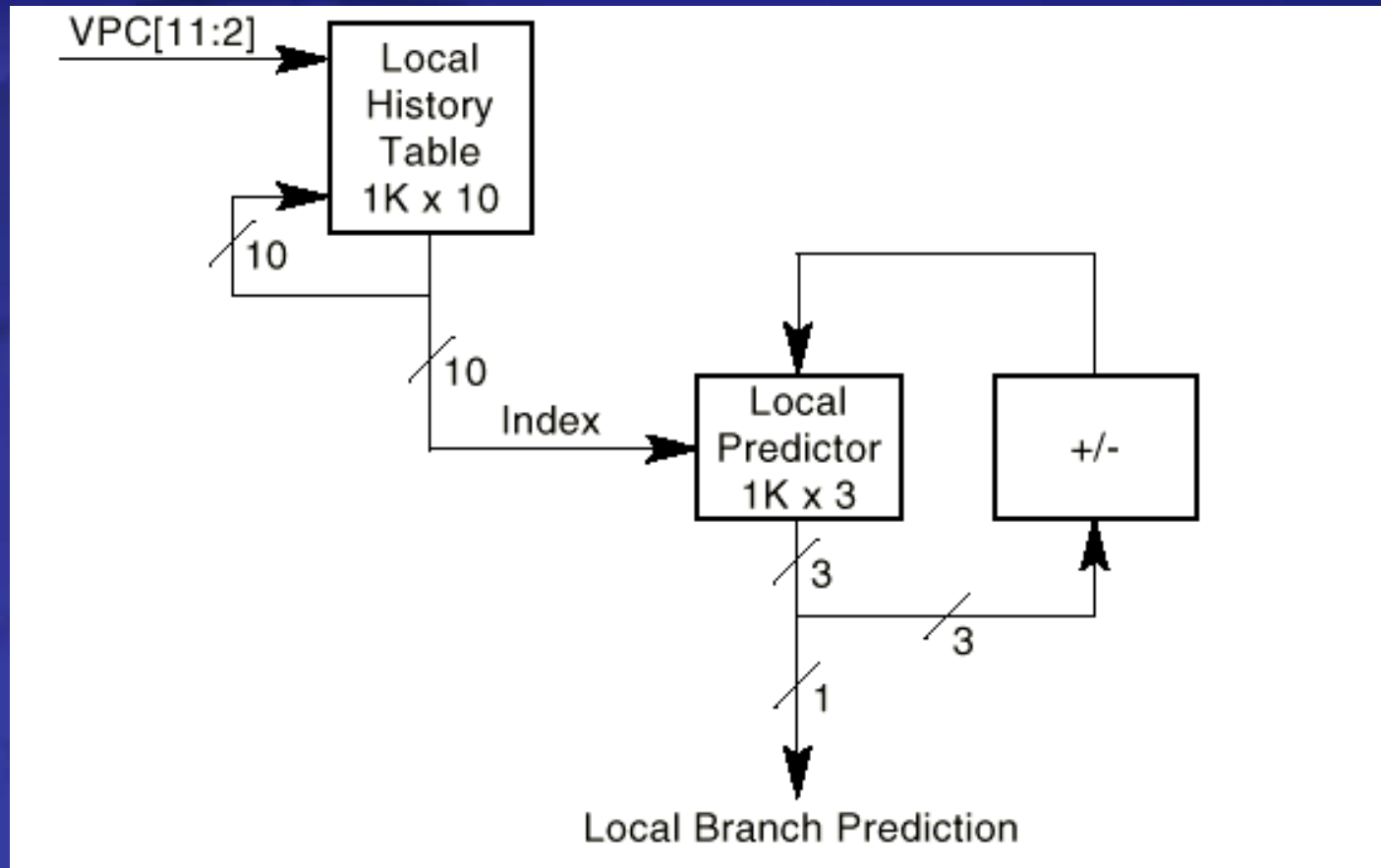
# Alpha 21264 Tournament Branch Prediction



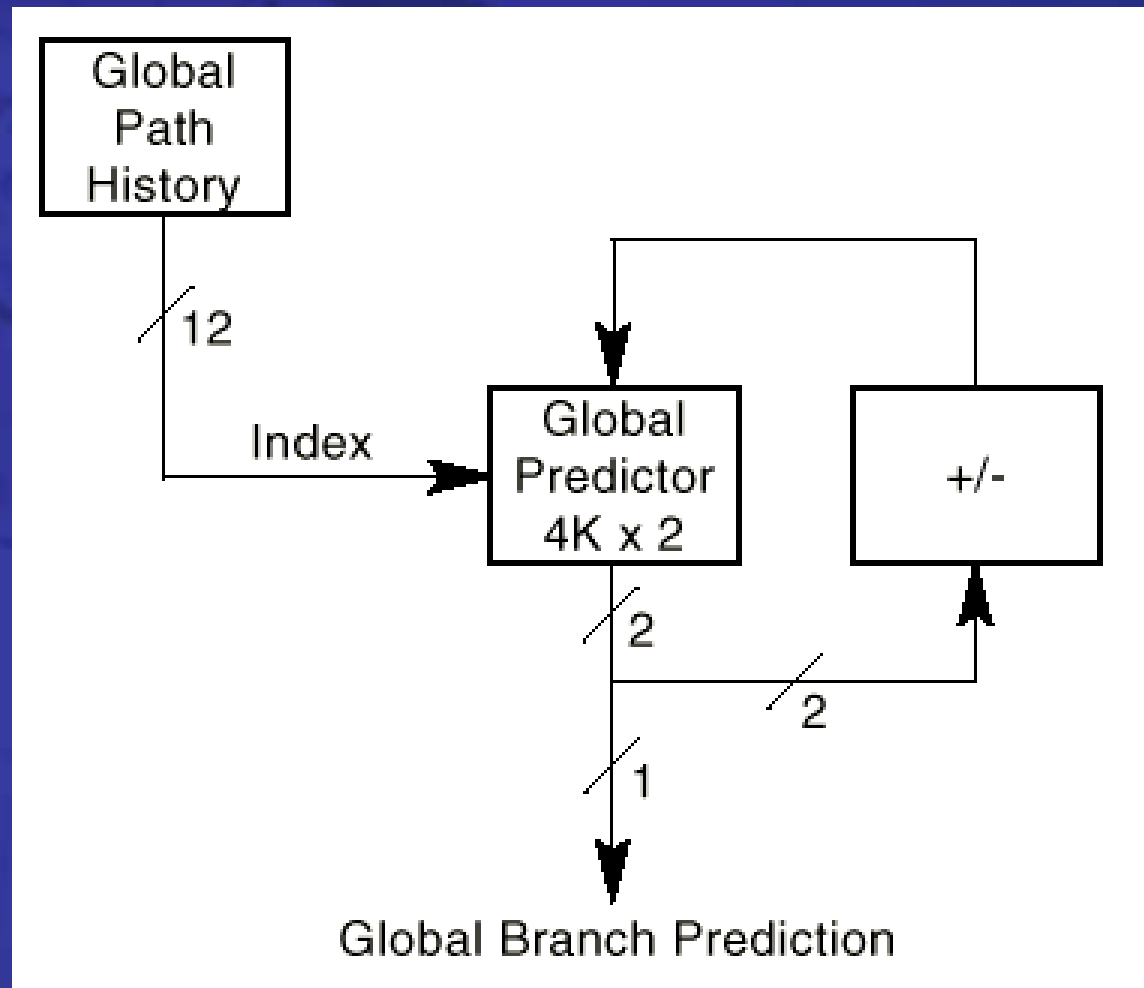
# Branch Prediction: Motivation

- Highly pipelined design introduces additional misprediction penalty cycles .
- Instructions typically spend 4-5 cycles in instruction queue -- more delay!
- Minimum penalty: 7 cycles
- Average penalty: > 11 cycles
- 2-cycle instruction cache adds another cycle to branch misprediction penalty -- sounds like a lot, but only 10%

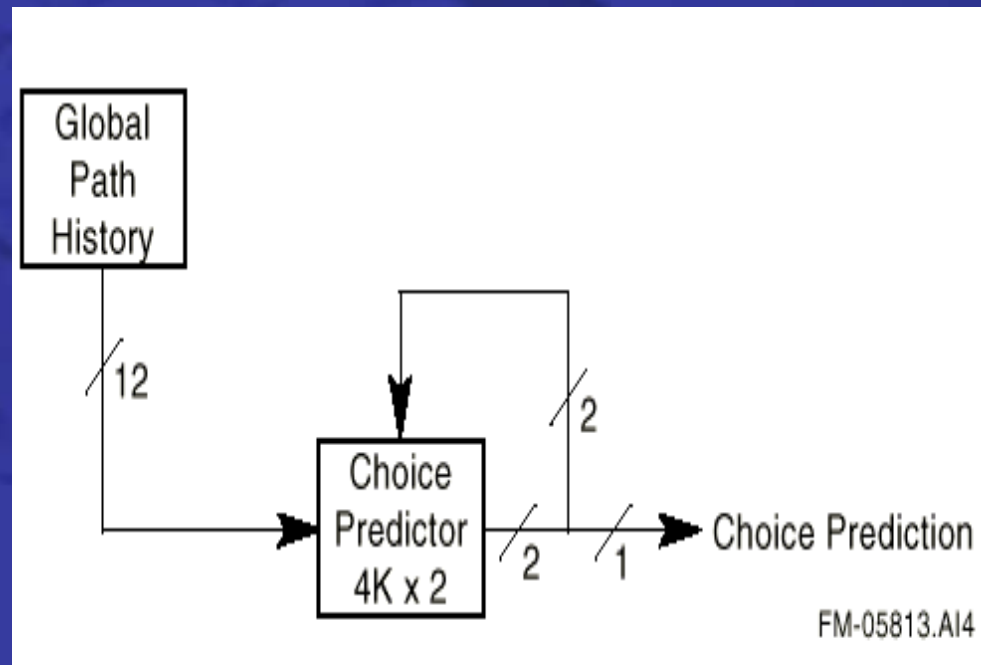
# Local Predictor



# Global Predictor



# Choice Predictor



## Other Branch Prediction Stuff

- 32 entry return-address stack to predict target of subroutine returns. Situated near the instruction cache and accessed in parallel.
- 35K bits of storage for branch history information = 2% of die area.
- 48K bits of target address information stored in cache .
- 7-10 mispredictions per 1000 instructions on SPECint95 \* 11 cycles per misprediction = .1 CPI loss.

# Branch Prediction Problems

- Instructions must always be grouped in fours, with predicted-taken branch in slot four, and branch targets in slot one, in order to issue four instructions on a branch.
- Context switching -- every time the CPU does a context switch, branch prediction tables are reset.

# Instruction-Stream Translation Buffer(ITB)

- 128-entry, fully associative ITB.
- Store recently used instruction-stream(Istream) address translations and page protection information.
- ITB is accessed only for the Istream references that miss in the Icache.

# Instruction Fetch Logic

- Up to four aligned instructions are fetched from the Icache, in program order.
- The branch prediction tables are also accessed in this cycle( instruction fetch cycle or Stage 0).
- The branch predictor uses tables and a branch history algorithm to predict a branch instruction target address for one branch or memory format.
- Branch prediction and line prediction bits accompany the four instructions.

# I-Cache

- Large 64 k byte, two-way set associative instruction cache.
- Each fetch block of four instructions includes a line and set prediction.
- Prediction indicates where to fetch the next block of instructions from, including which set should be used.
- Remove the bubble if the branch is predicted to be taken by the branch predictor.

## I-Cache(continued)

- On cache fills, the line predictor value at each fetch line is initialized with the index of the next sequential fetch line, and later retrained by the branch predictor if necessary.
- The line predictor does not train on every mispredict.
- The mispredict cost is typically a single cycle bubble.
- Line predictor is also trained for jumps that use direct register addressing.

## Instruction Fetch Logic(continued)

- In the slot stage(stage1), the branch predictor compares the next lcache index that it generates to the index that was generated by the line predictor.
- If there is a mismatch, the branch predictor wins and results in one bubble.
- The line predictor takes precedence over the branch predictor during memory format calls or jump.

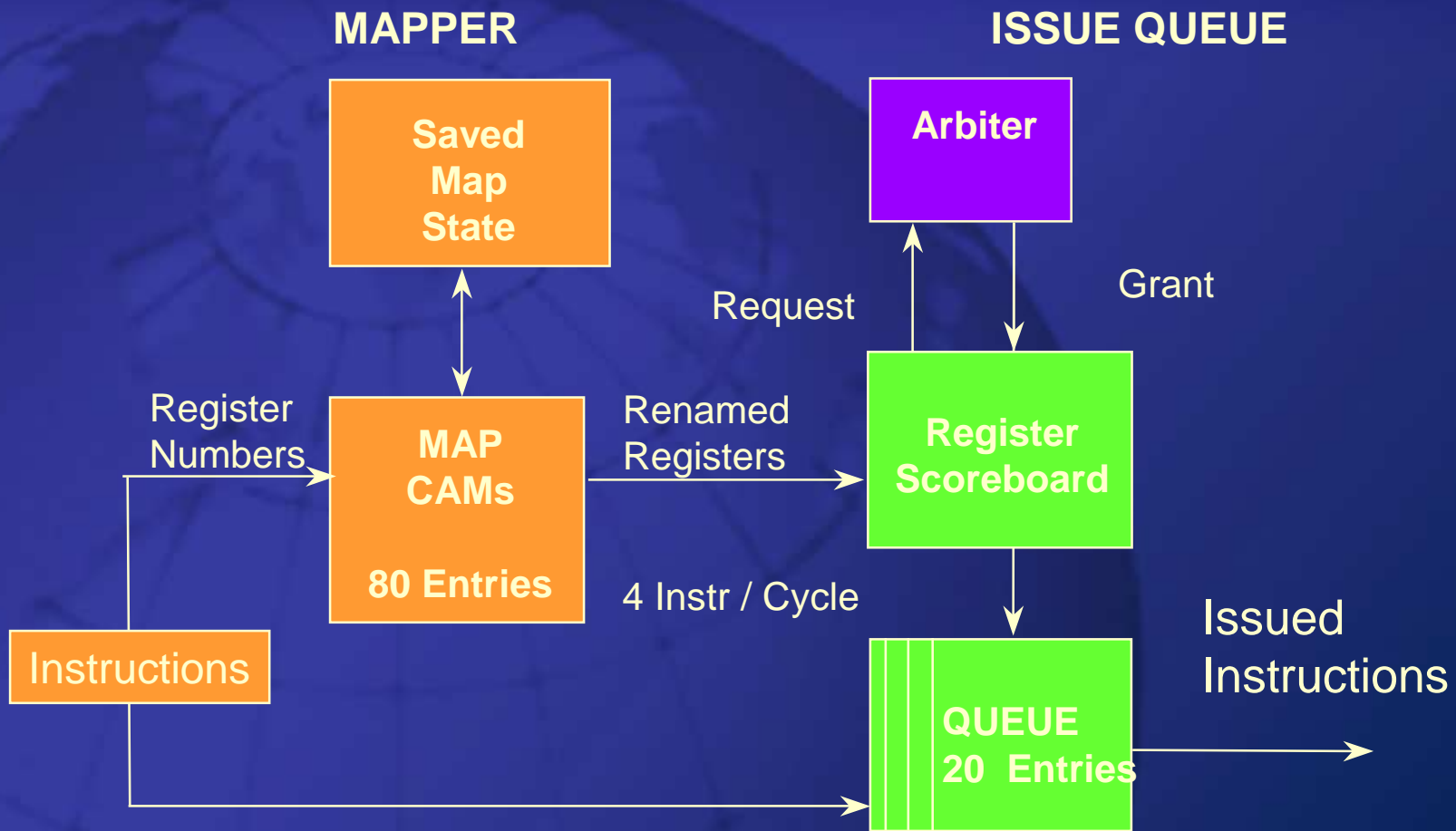
# Register Rename Maps

- In addition to the 64 architectural registers, up to 41 integer registers and 41 floating point registers.
- Eliminates write-after-read(WAR) and write-after-write(WAW) while preserving true read-after-write(RAW) dependence's.
- Provides a means of speculatively executing instructions.
- Each instruction is assigned a unique 8-bit number, called an *inum*, which is used to identify the instruction and it's program order.

## Register Rename Maps(continued)

- Map logic can process four instructions per cycle.
- Physical register is not returned to the free list until the instruction using it has been retired.
- If a branch mispredict or exception occurs, the map logic backs up the contents of the register rename maps to the state associated with the instruction that triggered the condition, and the prefetcher restarts at the appropriate VPC.
- Map logic can back up the contents of the maps to the state associated with any of 80 instructions in flight in a single cycle.

# Alpha 21264 Mapper and Queue Stages Block Diagram



# Integer Issue Queue(IQ)

- The 20-integer issue queue, associated with the integer execution units, issues instructions at the maximum rate of four per cycle.
- Each queue entry asserts four request signal(one for each sub-cluster( U0,U1,L0 &L1)).
- There are two arbiters-one for the upper sub-clusters and one for the lower sub-clusters.
- Some instructions like load and store can only go to lower sub-clusters and shift instructions can only go to the upper sub-clusters.

## Integer Issue Queue(continued)

- Other instructions , such as addition and logic operations, can execute in any sub-clusters and are statically assigned before being placed in the IQ.
- The IQ chooses between simultaneous requesters of a sub-cluster based on the age of the request- older request are given priority over newer requests.

## Floating-Point Issue Queue(FQ)

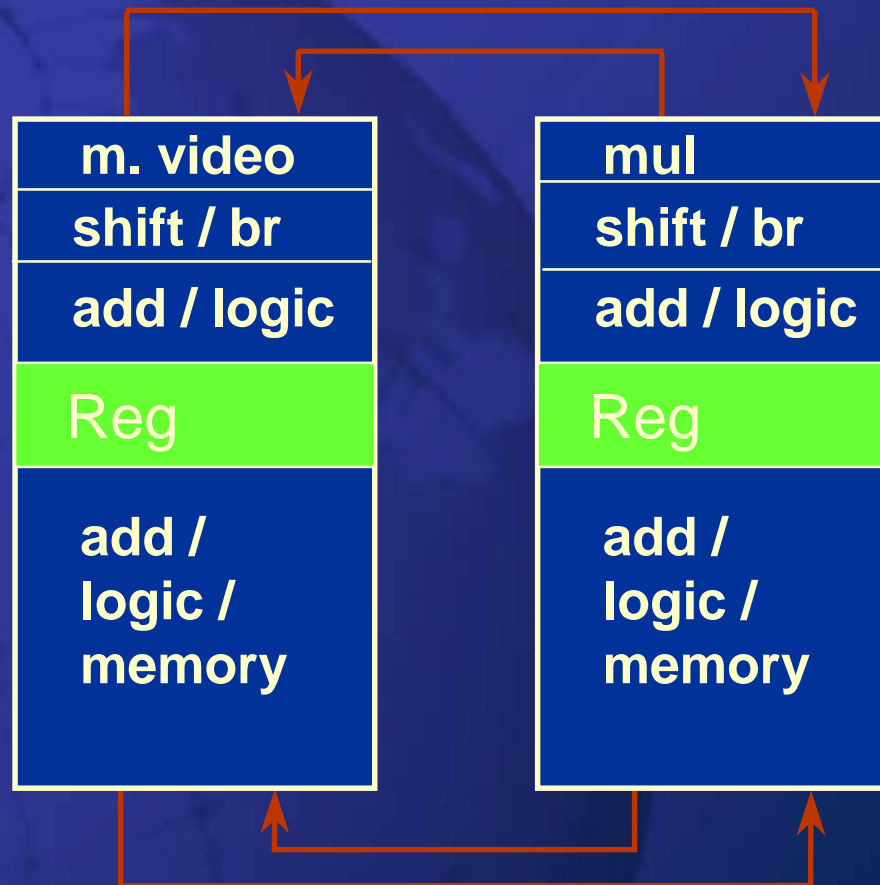
- The 15-entry FPQ is associated with the floating point execution units.
- Each queue entry has three request lines-one for the add pipeline,one for the multiply pipeline and one for the two store pipelines.
- There are three arbiters(one for each). The add and multiply arbiters pick one per cycle,while the store pipeline arbiter picks two requesters per cycle.
- The FQ arbiter picks between simultaneous requesters of a pipeline based on the age of the requests.

# Alpha 21264 Register and Execute Stages

## Floating Point Execution Units



## Integer Execution Units



# Exception and Interrupt Logic

- Two types of exception: faults and synchronous traps. Arithmetic exceptions are precise and are reported as synchronous traps.
- There are four types of interrupts: Two are hardware, one is software interrupt and the last is Asynchronous systems traps

# Retire Logic

- The Instruction box(Ibox) fetches instructions in program order,executes them out of order, and then retires them in order.
- The Ibox retire logic maintains the architectural state of the machine by retiring an instruction only if all previous instructions have executed without generating exceptions or branch mispredictions.
- Retiring an instruction commits the machine to any changes the instruction may have made to the software-visible state.

## Retire Logic(continued)

- The three software-visible states are:
  - Integer and floating-point registers
  - Memory
  - Internal processor registers
- The retire logic can sustain a maximum retire rate of 8 instructions per cycle, and can retire up to as many as 11 instructions in a single cycle.