Case Studies: Mainstream Superscalar Architectures--The PowerPC 620 and Intel P6

55:132/22C:160
Spring 2011

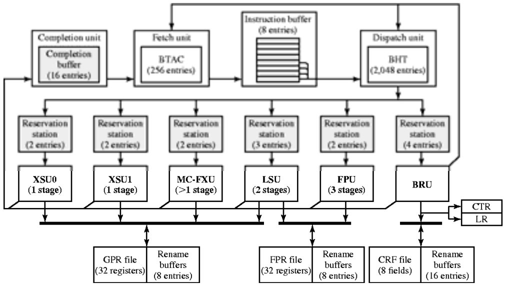
PowerPC 620 Case Study

- First-generation out-of-order processor
- Developed as part of Apple-IBM-Motorola alliance
- Aggressive goals, targets
- Interesting microarchitectural features
- Hopelessly delayed
- Led to future, successful designs

IBM/Motorola/Apple Alliance

- **Alliance begun in 1991 with a joint design center (Somerset) in Austin**
  - **Ambitious objective: unseat Intel on the desktop**
  - **Delays, conflicts, politics…hasn't happened, alliance largely dissolved today**
- **PowerPC 601**
  - Quick design based on RSC compatible with POWER and PowerPC
- **PowerPC 603**
  - Low power implementation designed for small uniprocessor systems
  - 5 FUs: branch, integer, system, load/store, FP
- **PowerPC 604**
  - 4-wide machine
  - 6 FUs, each with 2-entry RS
- **PowerPC 620**
  - First 64-bit machine, also 4-wide
  - Same 6 FUs as 604
  - Next slide, also chapter 5 in the textbook
- **PowerPC G3, G4**
  - Newer derivatives of the PowerPC 603 (3-issue, in-order)
  - Added Altivec multimedia extensions

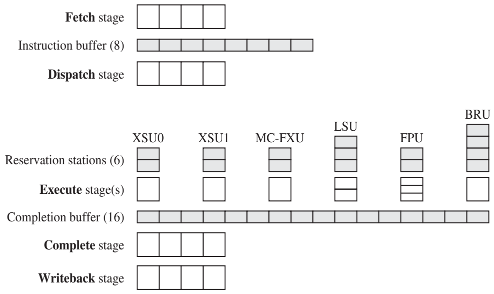
PowerPC 620



Completion unit
Completion buffer (16 entries)
Fetch unit
BTAC (256 entries)
Instruction buffer (8 entries)
Dispatch unit
BHT (2,048 entries)
Reservation station (2 entries)
Reservation station (2 entries)
Reservation station (2 entries)
Reservation station (3 entries)
Reservation station (2 entries)
Reservation station (4 entries)
XSU0 (1 stage)
XSU1 (1 stage)
MC-FXU (>1 stage)
LSU (2 stages)
FPU (3 stages)
BRU
CTR
LR
GPR file (32 registers)
Rename buffers (8 entries)
FPR file (32 registers)
Rename buffers (8 entries)
CRF file (8 fields)
Rename buffers (16 entries)

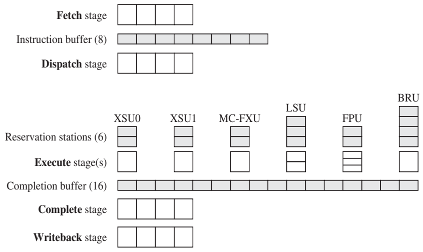
- **PowerPC 620**
  - **Joint IBM/Apple/Motorola design**
  - **Aggressively out-of-order, weak memory order, 64 bits**
- **Hopelessly delayed, very few shipped, but influenced later designs**

PowerPC 620 Pipeline



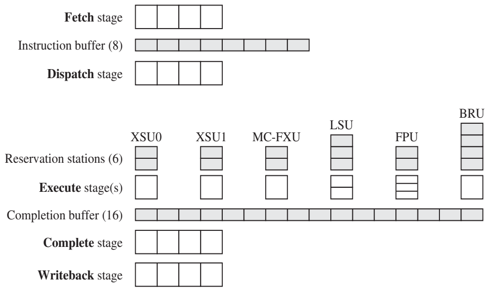
- Fetch stage
  - 4-wide, BTAC simple predictor
- Instruction Buffer
  - Decouples fetch from dispatch stalls
  - Holds up to 8 instructions

PowerPC 620 Pipeline



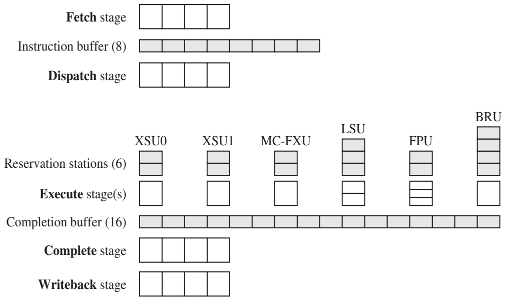
- Dispatch Stage
  - Rename
  - Allocate: rename buffer, completion buffer
  - Dispatch to reservation station
  - Branches: resolve (if operands avail.) or predict with BHT
- Reservation Stations
  - 2 to 4 entries per functional unit, depending on type
  - RS holds instruction payload, operands

PowerPC 620 Pipeline



- Execute Stage
  - Six functional units
  - Execute, bypass to waiting RS entries, write rename buffer
- Completion Buffer
  - Sixteen entries, holds instruction state until in-order completion

PowerPC 620 Pipeline



- Complete Stage
  - Maintains precise exceptions by buffering out-of-order instructions
  - 4-wide
- Writeback Stage
  - In-order writeback: results copied from rename buffer to architected register file

Benchmark Performance

Benchmarks	Dynamic Instructions	Execution Cycles	IPC
compress	6,884,247	6,062,494	1.14
Eqntott	3,147,233	2,188,331	1.44
espresso	4,615,085	3,412,653	1.35
Li	3,376,415	3,399,293	0.99
alvinn	4,861,138	2,744,098	1.77
hydro2d	4,114,602	4,293,230	0.96
tomcatv	6,858,619	6,494,912	1.06

Branch Prediction

- Two-phase branch prediction
  - BTAC
    - Holds targets of taken branches only
      - On miss, fetch sequential (not-taken) path
    - Accessed in single cycle in fetch stage
    - Generates fetch address for next cycle
    - 256 entries, 2-way set-associative
  - BHT
    - Accessed in dispatch stage
    - 2048 entries of 2-bit counters (bimodal)
  - Also attempts to resolve branches at dispatch
- Interactions
  - {BTAC right, wrong} x {BHT right, wrong} = 4 cases
  - BHT overrides BTAC

Branch Prediction Accuracy

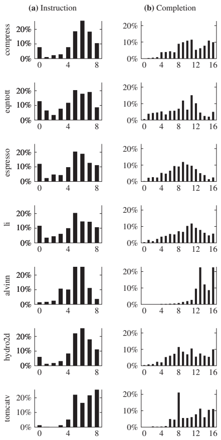
	compress	eqntott	espresso	li	alvinn	hydro2d	tomcatv
BranchResolution							
Not Taken	40.35%	31.84%	40.05%	33.09%	6.38%	17.51%	6.12%
Taken	59.65%	68.16%	59.95%	66.91%	93.62%	82.49%	93.88%
BTACPrediction							
Correct	84.10%	82.64%	81.99%	74.70%	94.49%	88.31%	93.31%
Incorrect	15.90%	17.36%	18.01%	25.30%	5.51%	11.69%	6.69%
BHT Prediction							
Resolved	19.71%	18.30%	17.09%	28.83%	17.49%	26.18%	45.39%
Correct	68.86%	72.16%	72.27%	62.45%	81.58%	68.00%	52.56%
Incorrect	11.43%	9.54%	10.64%	8.72%	0.92%	5.82%	2.05%
BTAC Incorrect and BHT Correct	0.01%	0.79%	1.13%	7.78%	0.07%	0.19%	0.00%
BTAC Correct and BHT Incorrect	0.00%	0.12%	0.37%	0.26%	0.00%	0.08%	0.00%
Overall Branch Prediction Accuracy	**88.57%**	**90.46%**	**89.36%**	**91.28%**	**99.07%**	**94.18%**	**97.95%**

Wasted Fetch Cycles

Benchmark	Misprediction	I-Cache Miss
compress	6.65%	0.01%
eqntott	11.78%	0.08%
espresso	10.84%	0.52%
li	8.92%	0.09%
alvinn	0.39%	0.02%
hydro2d	5.24%	0.12%
tomcatv	0.68%	0.01%

Buffer Utilization

- Instruction buffer
  - Decouples fetch/dispatch
- Completion buffer
  - Supports OOO execution



Dispatch Stalls

Frequency of dispatch stall cycles.
Sources of Dispatch Stalls

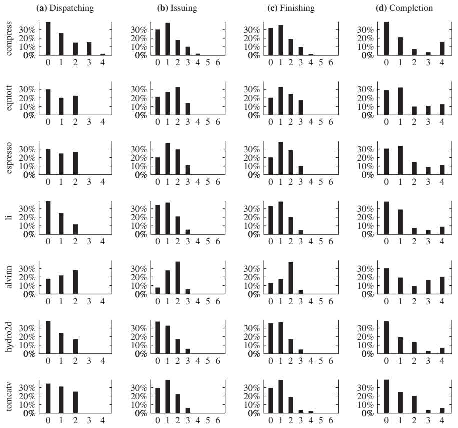
	compress	eqntott	espresso	li	alvinn	hydro2d	tomcatv
Serialization	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Move to special register constraint	0.00%	4.49%	0.94%	3.44%	0.00%	0.95%	0.08%
Read port saturation	0.26%	0.00%	0.02%	0.00%	0.32%	2.23%	6.73%
Reservation station saturation	36.07%	22.36%	31.50%	34.40%	22.81%	42.70%	36.51%
Rename buffer saturation	24.06%	7.60%	13.93%	17.26%	1.36%	16.98%	34.13%
Completion buffer saturation	5.54%	3.64%	2.02%	4.27%	21.12%	7.80%	9.03%
Another to same unit	9.72%	20.51%	18.31%	10.57%	24.30%	12.01%	7.17%
No dispatch stalls	24.35%	41.40%	33.28%	30.06%	30.09%	17.33%	6.35%

Issue Stalls

Frequency of issue stall cycles.
Sources of Issue Stalls

	compress	eqntott	espresso	li	alvinn	hydro2d	tomcatv
Out of order disallowed	0.00%	0.00%	0.00%	0.00%	0.72%	11.03%	1.53%
Serialization	1.69%	1.81%	3.21%	10.81%	0.03%	4.47%	0.01%
Waiting for source	21.97%	29.30%	37.79%	32.03%	17.74%	22.71%	3.52%
Waiting for execution unit	13.67%	3.28%	7.06%	11.01%	2.81%	1.50%	1.30%
No issue stalls	62.67%	65.61%	51.94%	46.15%	78.70%	60.29%	93.64%

Parallelism Achieved



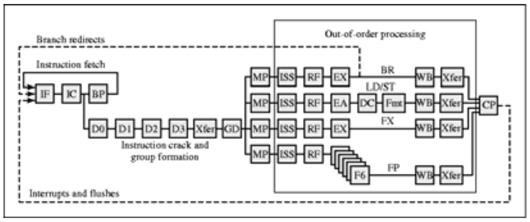
Summary of PowerPC 620

- First-generation OOO part
- Aggressive goals, poor execution
- Interesting contributions
  - Two-phase branch prediction (also in 604)
  - Short pipeline
  - Weak ordering of memory references
- PowerPC evolution
  - 1998: Power3 (630FP)
  - 2001: Power4
  - 2004: Power5

620 vs. Power3 vs. Power4

Attribute	620	Power3	Power4
Frequency	172 MHz	450 MHz	1.3 GHz
Pipeline depth	5+	5+	15+
Branch predictor	Bimodal BHT + BTAC	Same	3x16 1b combining
Fetch/issue/completion width	4/6/4	4/8/4	4/8/5
Rename/physical registers	8 Int, 8 FP	16 Int, 24 FP	80 Int, 72 FP
In-flight instructions	16	32	Up to 100
FP Units	1	2	2
Load/store units	1	2	2
Instruction Cache	32K 8w SA	32K 128w SA	64K DM
Data Cache	32K 8w SA	64K 128w SA	32K 2w SA
L2/L3 size	4M	16M	1.5M/32M
L2 bandwidth	1GB/s	6.4GB/s	100+ GB/s
Store queue entries	6 x 8B	16 x 8B	12 x 64B
MSHRs	I:1/D:1	I:2/D:4	I:2/D:8
Hardware prefetch	None	4 streams	8 streams

IBM Power4



- **IBM POWER4, began shipping in 2001**
  - Deep pipeline: 15 stages minimum
  - Aggressive combining branch prediction
  - Over 100 instructions in flight, tracked in 20 groups of 5 in ROB
  - Aggressive memory hierarchy, memory bandwidth

Case Study: Intel P6 (Pentium Pro) Architecture

Pentium Pro Case Study

- **Microarchitecture**
  - – Order-3 Superscalar
  - – Out-of-Order execution
  - – Speculative execution
  - – In-order completion
- **Design Methodology**
- **Performance Analysis**

Goals of P6 Microarchitecture

IA-32 Compliant

Performance (Frequency - IPC)

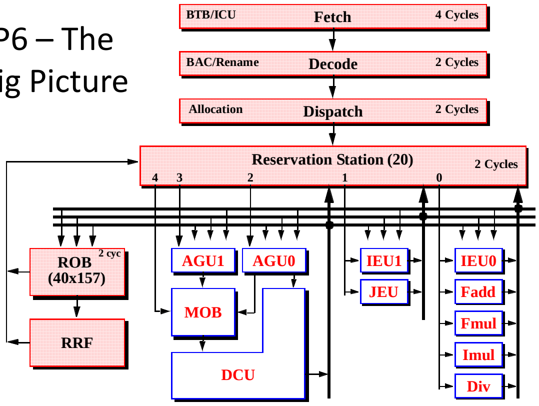
Validation

Die Size

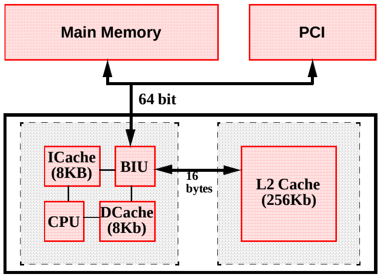
Schedule

Power

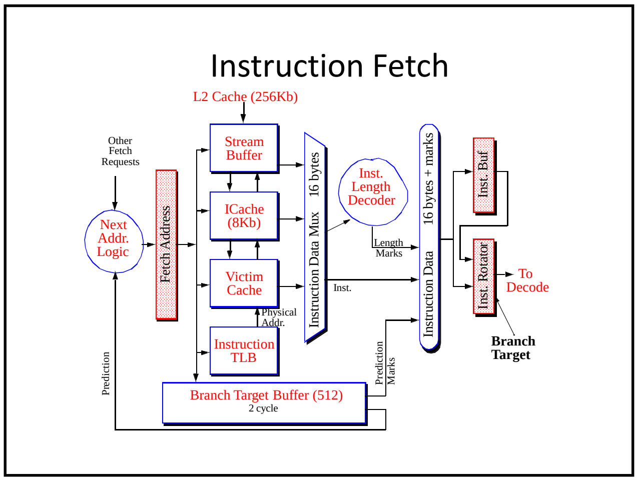
P6 – The Big Picture



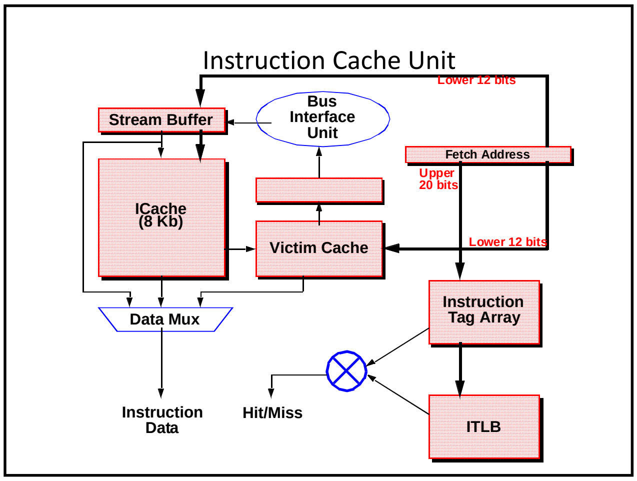
Memory Hierarchy



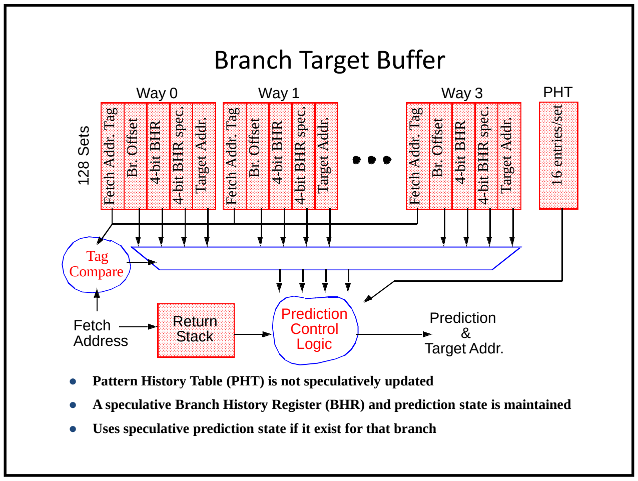
- **Level 1 instruction and data caches - 2 cycle access time**
- **Level 2 unified cache - 6 cycle access time**
- **Separate level 2 cache and memory address/data bus**
- **Level 2 cache fill policy - implications**



Instruction Fetch

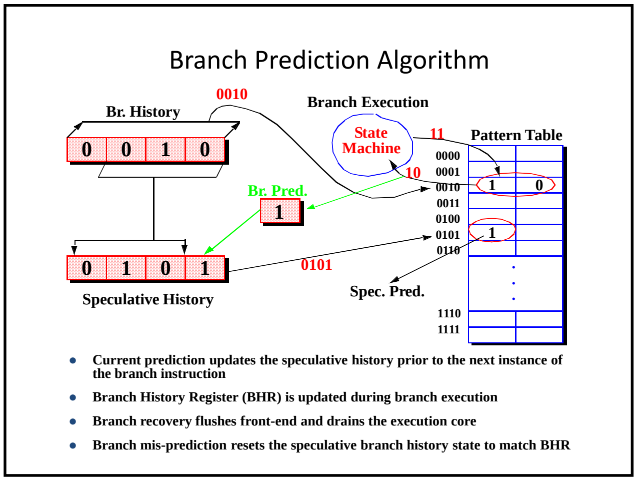


Instruction Cache Unit



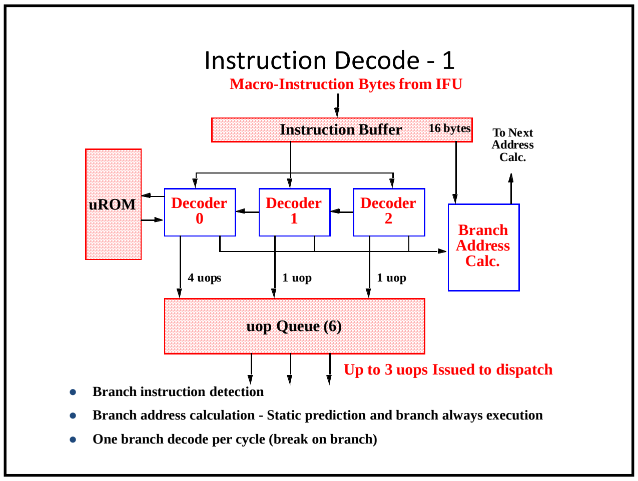
Branch Target Buffer

- Pattern History Table (PHT) is not speculatively updated
- A speculative Branch History Register (BHR) and prediction state is maintained
- Uses speculative prediction state if it exist for that branch



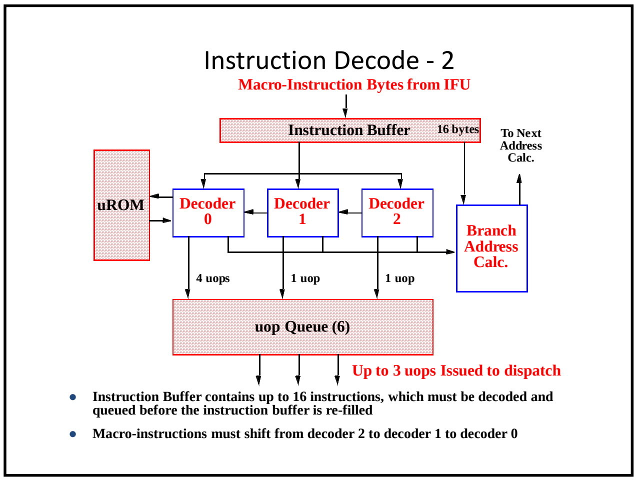
Branch Prediction Algorithm

- Current prediction updates the speculative history prior to the next instance of the branch instruction
- Branch History Register (BHR) is updated during branch execution
- Branch recovery flushes front-end and drains the execution core
- Branch mis-prediction resets the speculative branch history state to match BHR



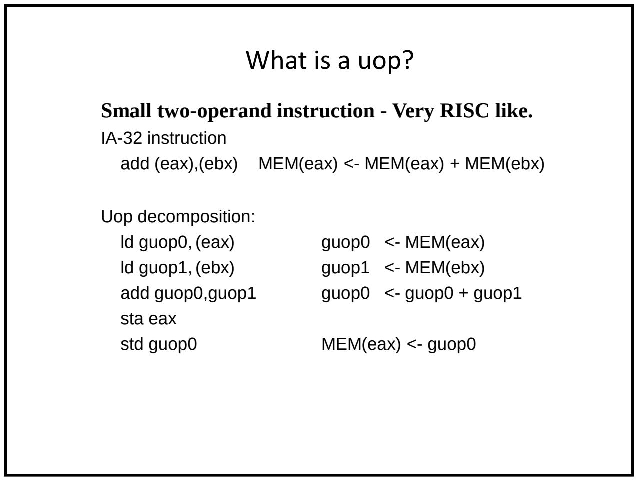
Instruction Decode - 1

- Branch instruction detection
- Branch address calculation - Static prediction and branch always execution
- One branch decode per cycle (break on branch)



Instruction Decode - 2

- Instruction Buffer contains up to 16 instructions, which must be decoded and queued before the instruction buffer is re-filled
- Macro-instructions must shift from decoder 2 to decoder 1 to decoder 0



What is a uop?

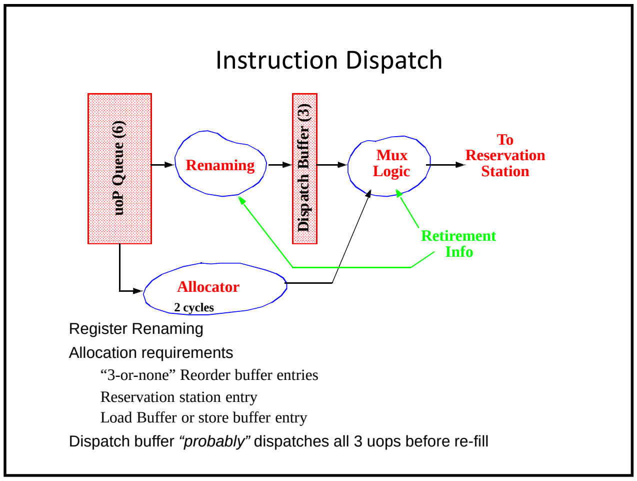
Small two-operand instruction - Very RISC like.

IA-32 instruction

add (eax),(ebx)   MEM(eax) <- MEM(eax) + MEM(ebx)

Uop decomposition:

ld guop0, (eax)	guop0 <- MEM(eax)
ld guop1, (ebx)	guop1 <- MEM(ebx)
add guop0,guop1	guop0 <- guop0 + guop1
sta eax	
std guop0	MEM(eax) <- guop0



Instruction Dispatch

Register Renaming

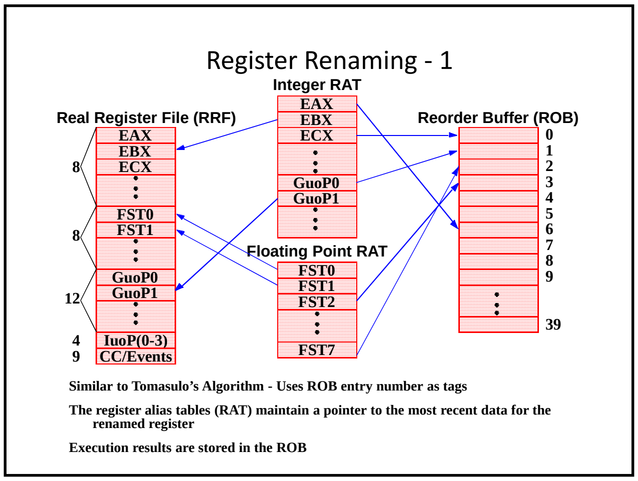
Allocation requirements

“3-or-none” Reorder buffer entries

Reservation station entry

Load Buffer or store buffer entry

Dispatch buffer *“probably”* dispatches all 3 uops before re-fill

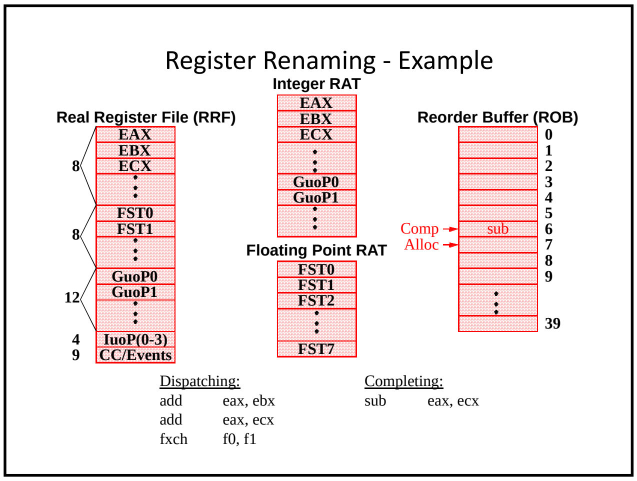


Register Renaming - 1

Similar to Tomasulo's Algorithm - Uses ROB entry number as tags

The register alias tables (RAT) maintain a pointer to the most recent data for the renamed register

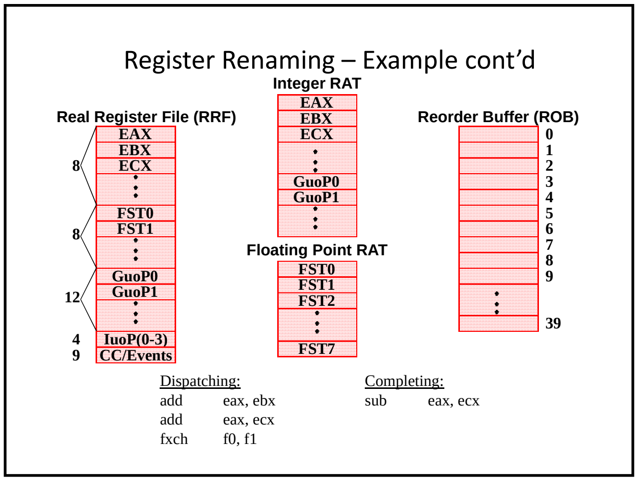
Execution results are stored in the ROB



Register Renaming - Example

Dispatching:
add eax, ebx
add eax, ecx
fxch f0, f1

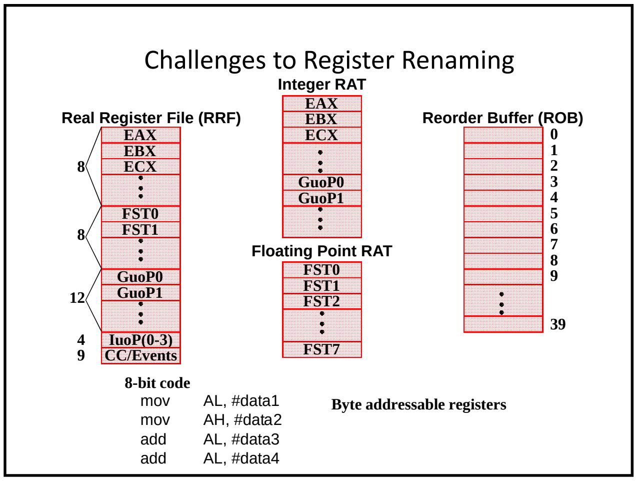
Completing:
sub eax, ecx



Register Renaming – Example cont'd

Dispatching:
add eax, ebx
add eax, ecx
fxch f0, f1

Completing:
sub eax, ecx



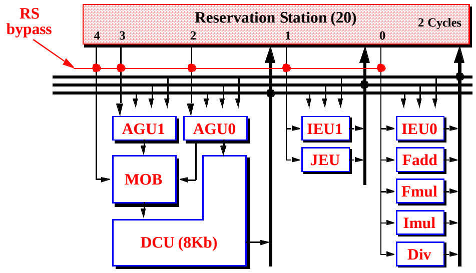
Challenges to Register Renaming

8-bit code

```
mov     AL, #data1
mov     AH, #data2
add     AL, #data3
add     AL, #data4
```

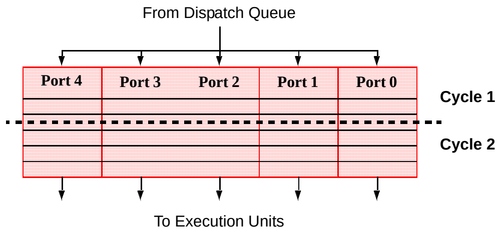
Byte addressable registers

Out-of-Order Execution Engine



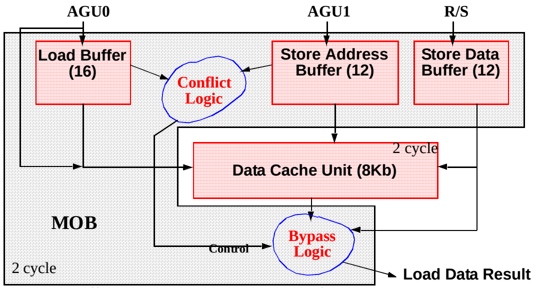
- **In-order branch issue and execution**
- **In-order load/store issue to address generation units**
- **Instruction execution and result bus scheduling**
- **Is the reservation station *"truly"* centralized & what is *"binding"*?**

Reservation Station



- Cycle 1
  - Order checking
  - Operand availability
- Cycle 2
  - Writeback bus scheduling

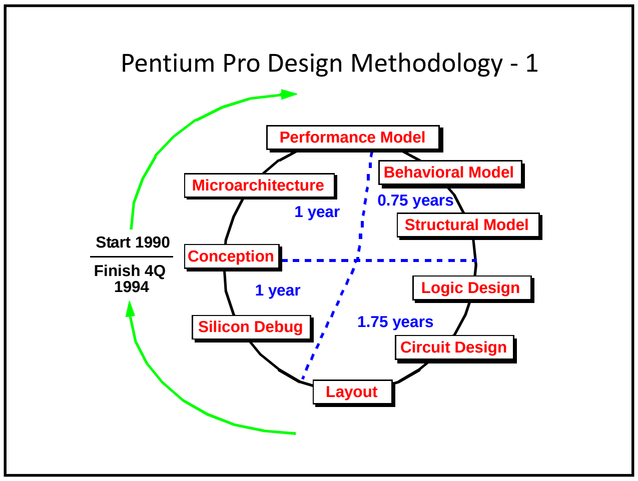
Memory Ordering Buffer (MOB)



- **Load buffer retains loads until completed, for coherency checking**
- **Store forwarding out of store buffers**
- **2 cycle latency through MOB**
- **"Store Coloring" - Load instructions are tagged by the last store**

Instruction Completion

- Handles all exception/interrupt/trap conditions
- Handles branch recovery
  - OOO core drains out right-path instructions, commits to RRF
  - In parallel, front end starts fetching from target/fall-through
  - However, no renaming is allowed until OOO core is drained
  - After draining is done, RAT is reset to point to RRF
  - Avoids checkpointing RAT, recovering to intermediate RAT state
- Commits execution results to the architectural state in-order
  - Retirement Register File (RRF)
  - Must handle hazards to RRF (writes/reads in same cycle)
  - Must handle hazards to RAT (writes/reads in same cycle)
- "Atomic" IA-32 instruction completion
  - uops are marked as 1st or last in sequence
  - exception/interrupt/trap boundary
- 2 cycle retirement



Pentium Pro Design Methodology - 1

Performance Model

Behavioral Model

Microarchitecture

1 year

0.75 years

Structural Model

Start 1990

Conception

Finish 4Q 1994

Logic Design

1 year

Silicon Debug

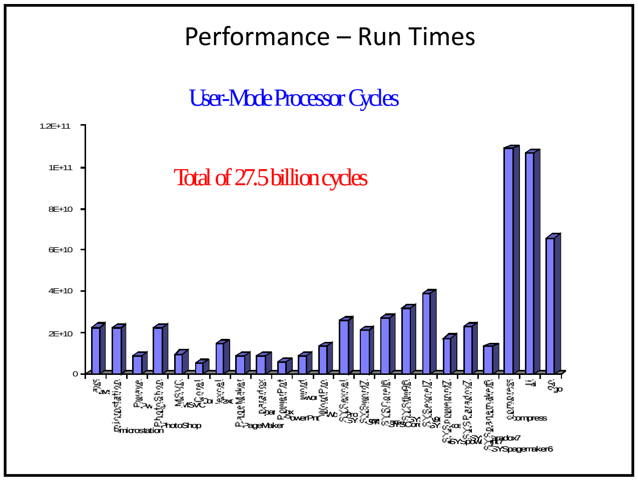
1.75 years

Circuit Design

Layout

Pentium Pro Performance Analysis

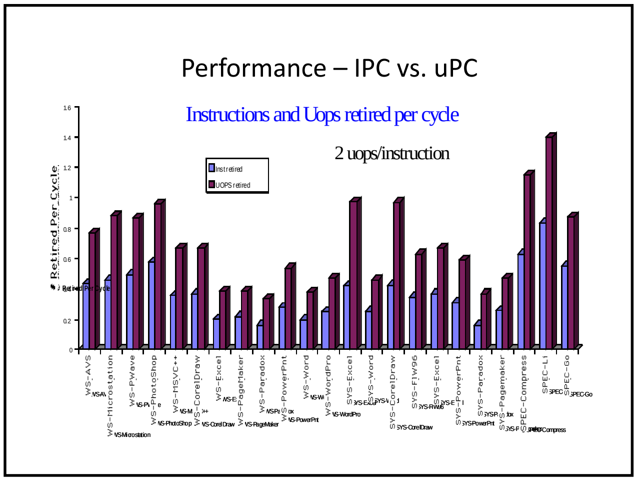
- **Observability**
  - On-chip event counters
  - Dynamic analysis
- **Benchmark Suite**
  - BAPco Sysmark32 - 32-bit Windows NT applications
  - Winstone97 - 32-bit Windows NT applications
  - Some SPEC95 benchmarks



Performance – Run Times

User-Mode Processor Cycles

Total of 27.5 billion cycles

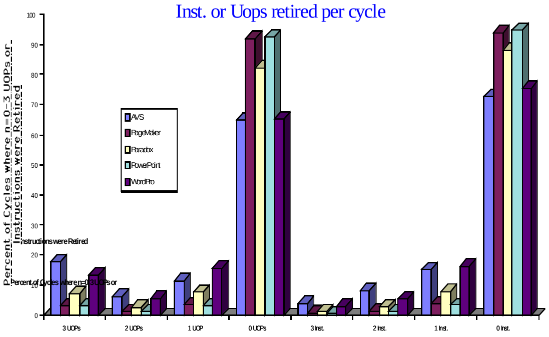


Performance – IPC vs. uPC

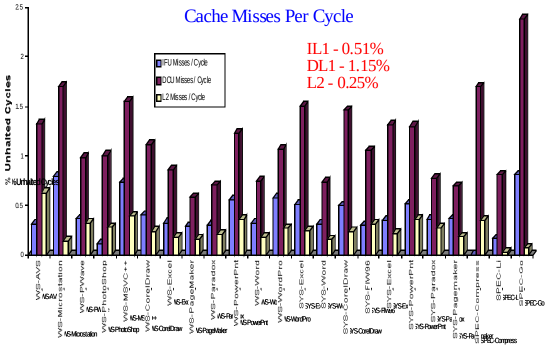
Instructions and Uops retired per cycle

2 uops/instruction

Performance – IPC vs. uPC

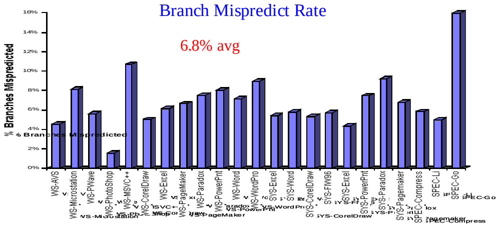

Inst. or Uops retired per cycle

Performance – Cache Misses

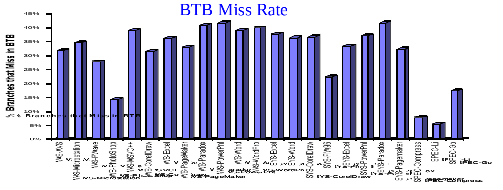

Cache Misses Per Cycle

IL1 - 0.51%
DL1 - 1.15%
L2 - 0.25%

Performance – Branch Prediction


Branch Mispredict Rate

6.8% avg


BTB Miss Rate