

UCI

University of California, Irvine

Intel Itanium Line Processor Efforts

Xiaobin Li

PASCAL
EECS Dept. UC, Irvine



- Intel Itanium Line Roadmap
- IA-64 Architecture
- Itanium Processor Microarchitecture
- Case Study of Exploiting TLP at VLIW

Intel Itanium Line Roadmap

Original

2001 Merced (Itanium)
 Mid-2002 McKinley (Itanium2)
 Mid-2003 Madison
 2004 Montecito

Now

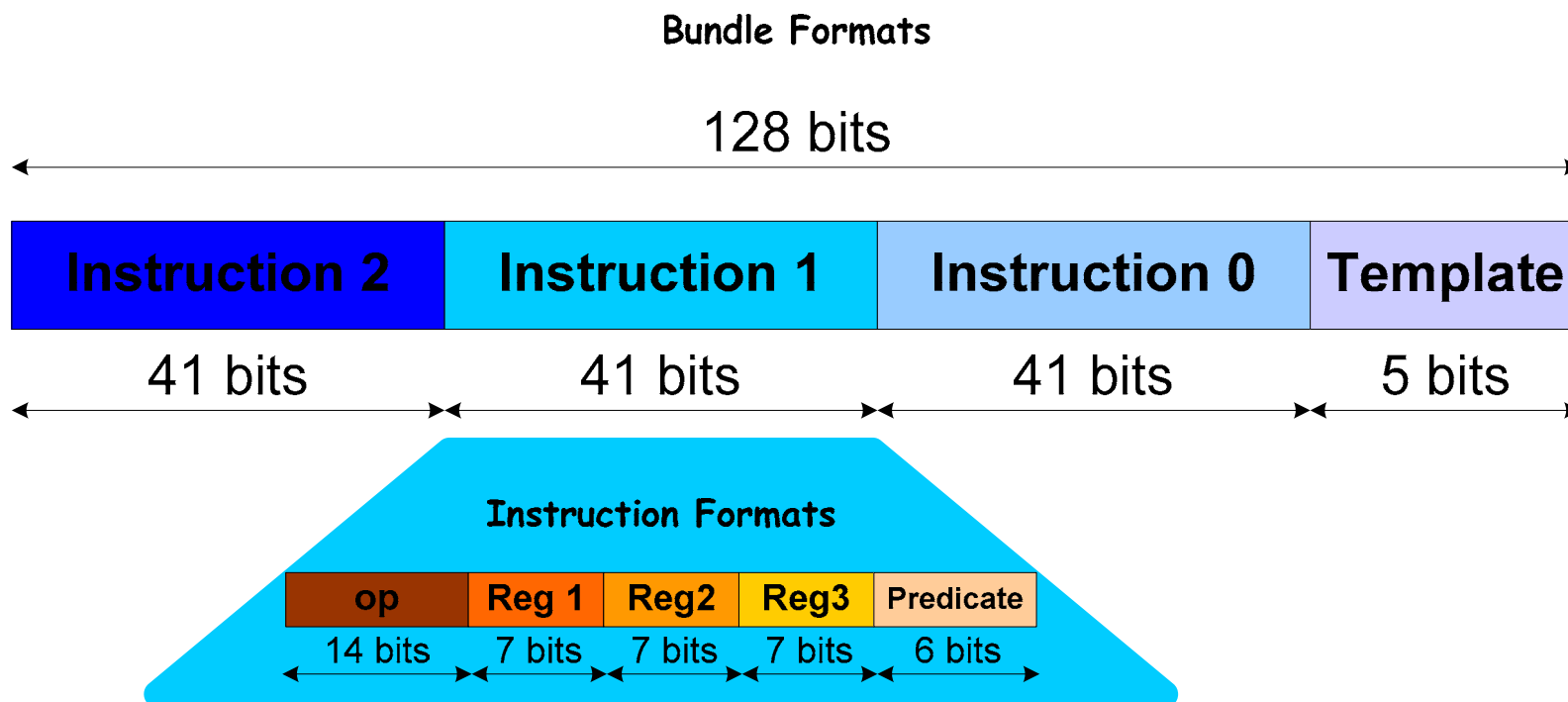
2001 Merced (Itanium)
 Mid-2002 McKinley (Itanium2)
 Mid-2003 Madison (1.5GHz, 130nm)
 2004 **Added!** Madison9M (9MB of L3 Cache)
 2005 **Delayed!** Montecito (**Dual-core**, 90nm)



EPIC (Explicitly Parallel Instruction Computing)

- An evolution of VLIW
- Three components:
 - 1. Explicit parallelism**
 - 2. Features to enhance ILP**
 - 3. Resources for parallel execution**

IA-64 Bundle and Instruction Formats



- Conditional execution of instructions
- Removes branches, results in larger basic blocks
- Reduces associated mispredict penalties

```
if (r1 == r2)
    r9 = r10 - r11;
else
    r5 = r6 + r7;
```

Branch prediction with speculative execution

- (if r1 == r2), branch predicted
- Speculative instructions executed
- Branch resolved (misprediction)
- Speculation instruction squashed
- Correct instruction executed

Predication code

```
cmp.eq p1, p2 = r1, r2;;
(p1) sub r9 = r10, r11
(p2) add r5 = r6, r7
```

Predication execution

- Compare (r1 == r2)
- Both paths executed simultaneously

Control Speculation

- Code motion: moving a load (and its uses) past a branch
- Hardware exception deferral

Traditional scheduling

```

instr A
instr B
    ...
br
-----
ld8 r1 = [r2]
use r1
    ...
    
```

IA-64 scheduling

```

ld8.s r1 = [r2]
use r1

instr A
instr B
    ...
br
-----
chk.s
    ...
    
```

Data Speculation

- Code motion: moving a load advance to its preceding stores, even if the load and stores may alias
- Hardware support: advanced load address table (ALAT)

Conservative Scheduling

```

instr A
instr B
...
store
ld8 r1 = [r2]
use r1
    
```

IA-64 Scheduling

```

ld8.a r1 = [r2]
use r1

instr A
instr B
...
store
chk.a
    
```

ALAT Organization

dest reg#	addr	size
r1	[r2]	8
...

stores CAM
on addr
field

Software Pipelining

- Software pipelining for loop-level parallelism: prologue, kernel, epilogue
- Reduced software pipelining overheads:
 - * **Register rotation** (register rename base (RRB) register)
=>provides renaming of intermediate results from previous iterations, eliminating register copy operations in the loop body;
 - * **Stage predicate** =>control execution of instruction in the stage;
 - * **Loop count** (LC) => determines whether the kernel of the loop will be executed or not;
 - * **Epilogue count** (EC) =>schedule the number of the loop iterations required to drain the software pipeline

Register Rotation

Register Set	Static	Rotating
General Registers	0-31	32-127
FP Registers	0-31	32-127
Predicate Registers	0-15	16-63

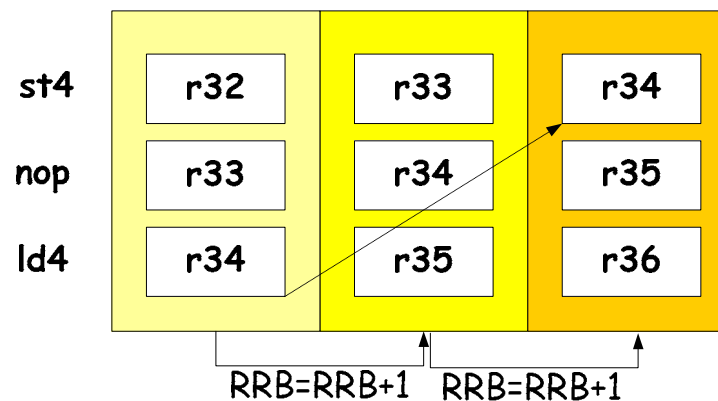
In the range of rotating: $\text{Reg\#} = \text{RRB} + \text{Reg\#}$

```
Loop:ld4 r34 = [r10],4
```

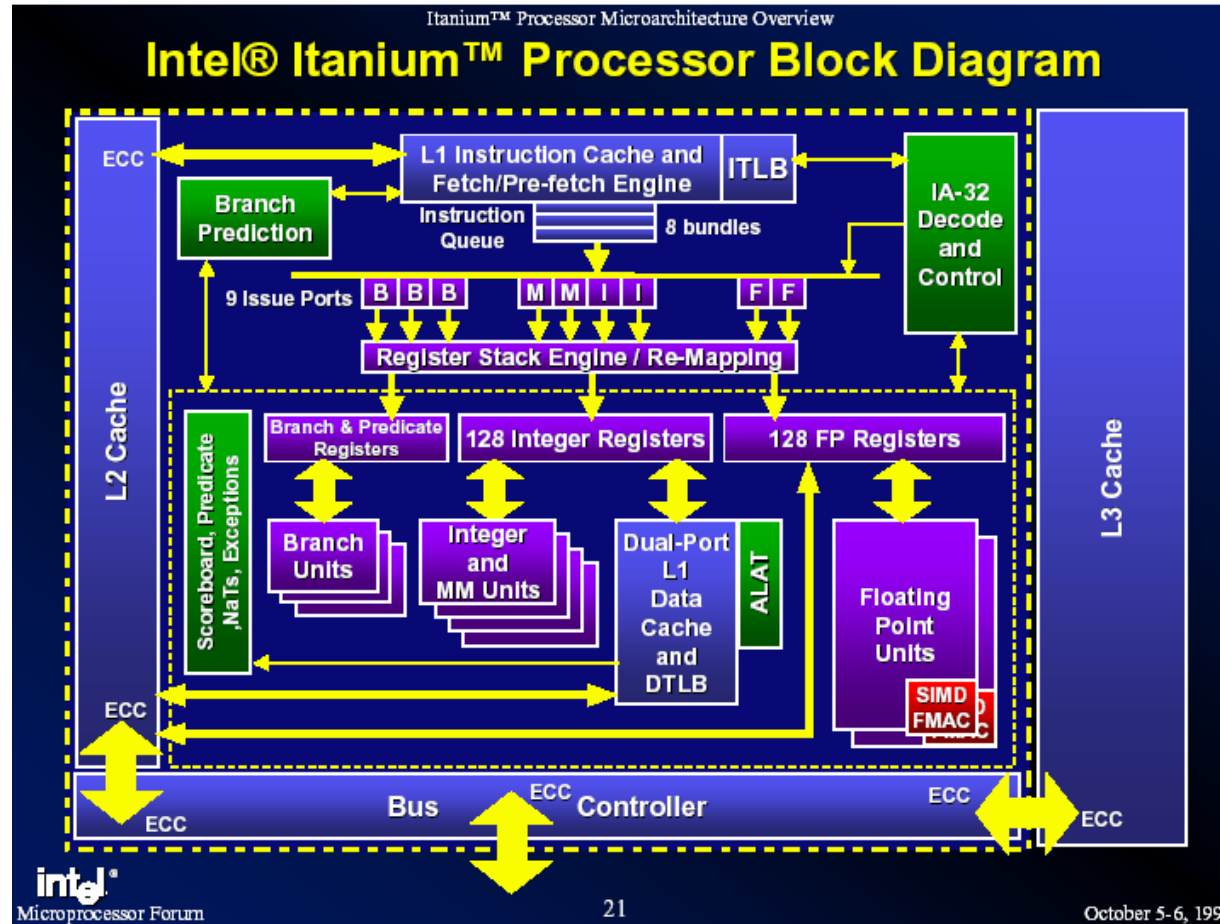
```
    nop
```

```
    st4 [r11] = r32,4
```

```
    swp_branch Loop; ;
```

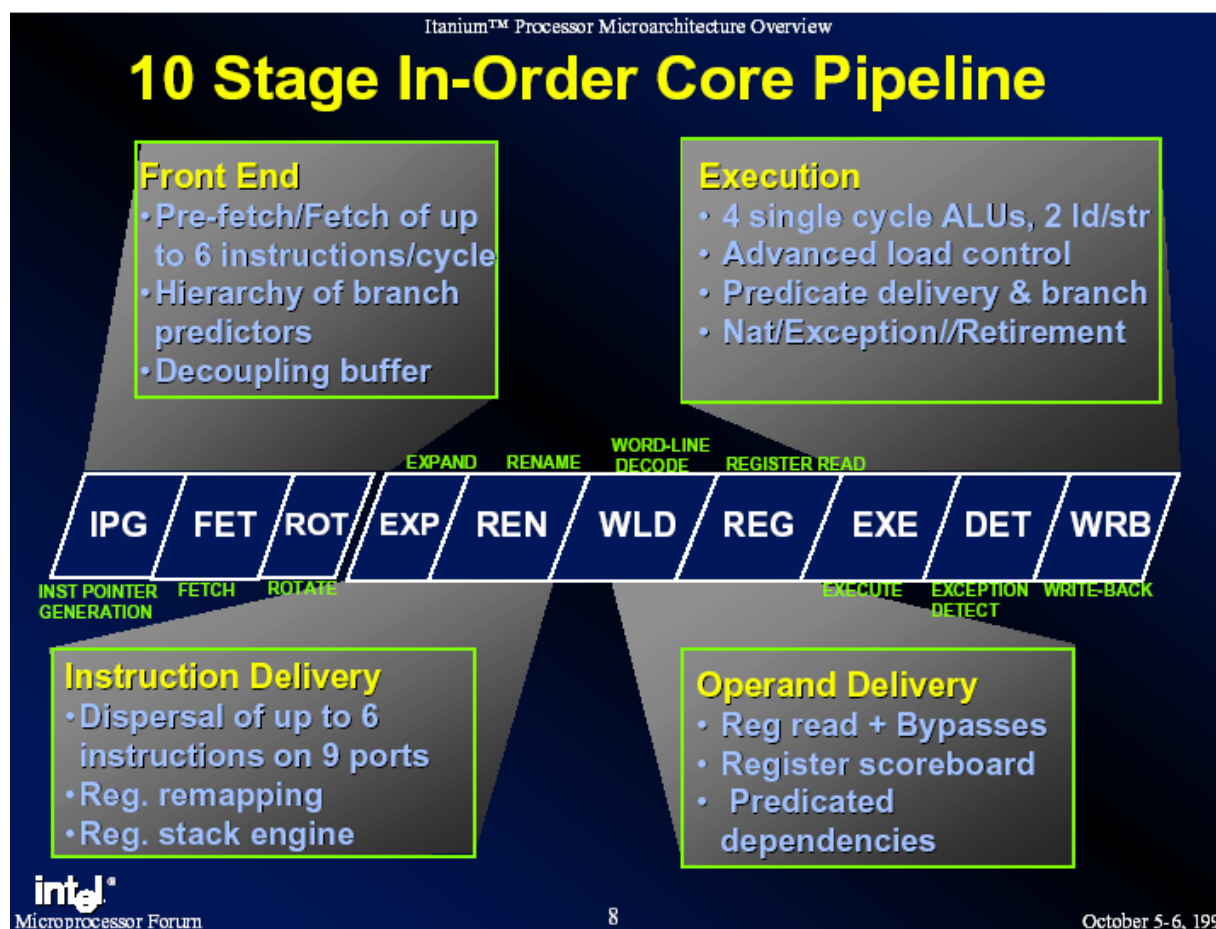


Itanium Processor Microarchitecture



From: "Intel(R) Itanium(TM) Processor Microarchitecture Overview"

Itanium Processor Core Pipeline



Thread-Level Parallelism

- ILP: VLIW, predication, control speculation, data speculation, software pipelining
- TLP: CMP, multi-threading, SMT

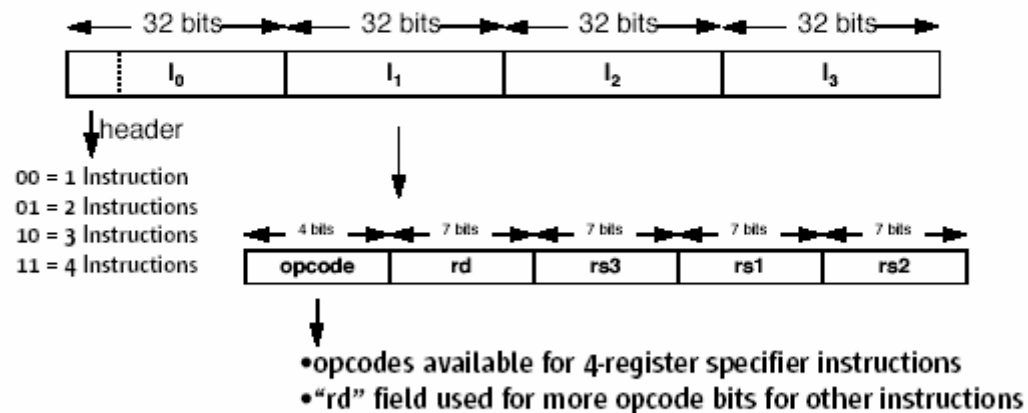
	Superscalar	VLIW/EPIC
CMP	✓	✓
MT	✓	
SMT	✓	

Dual-Core VLIW Processor Case Study

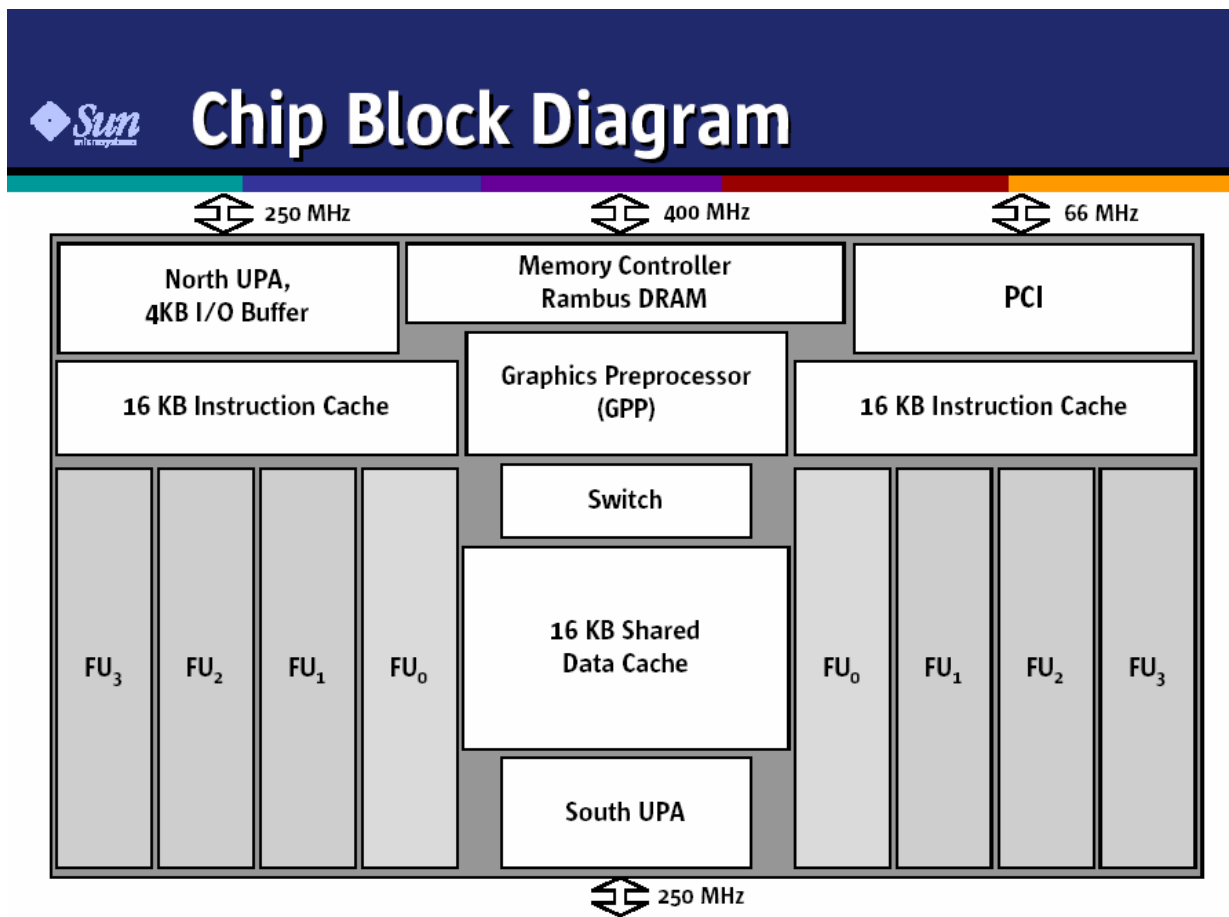




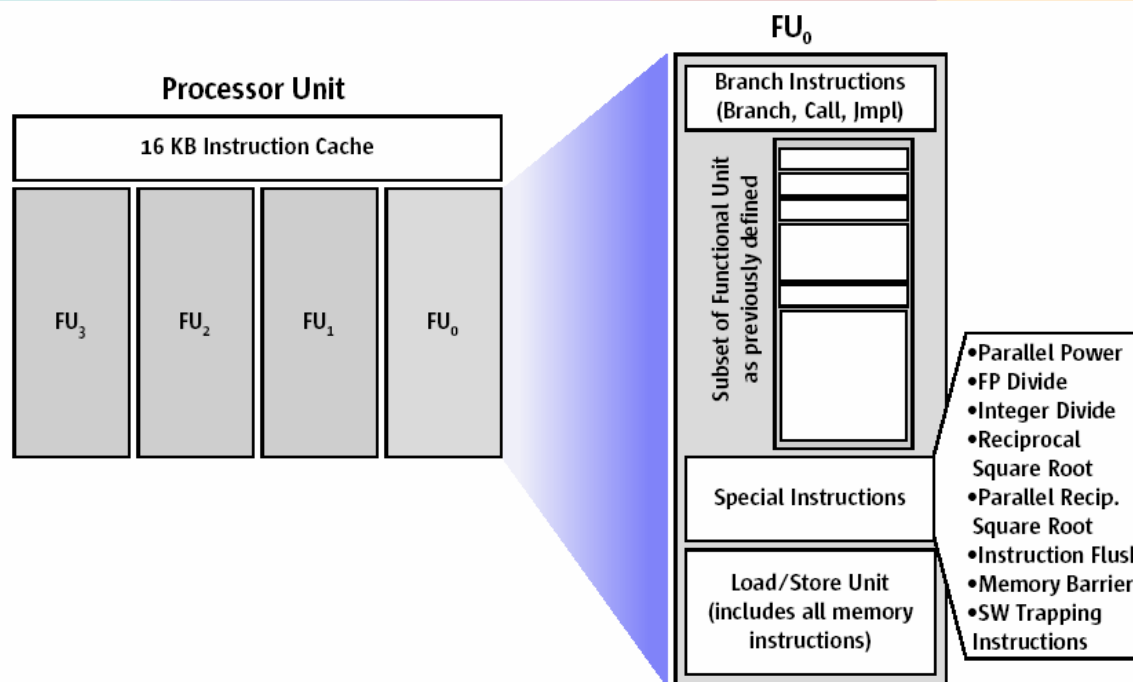
128-bit VLIW Instruction Packets



MAJC 5200 Chip

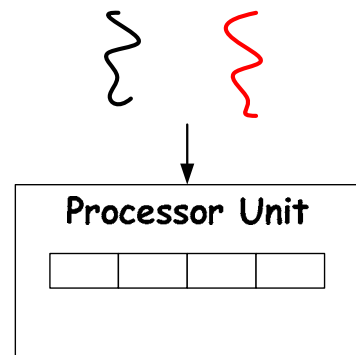


MAJC 5200 Processor Unit



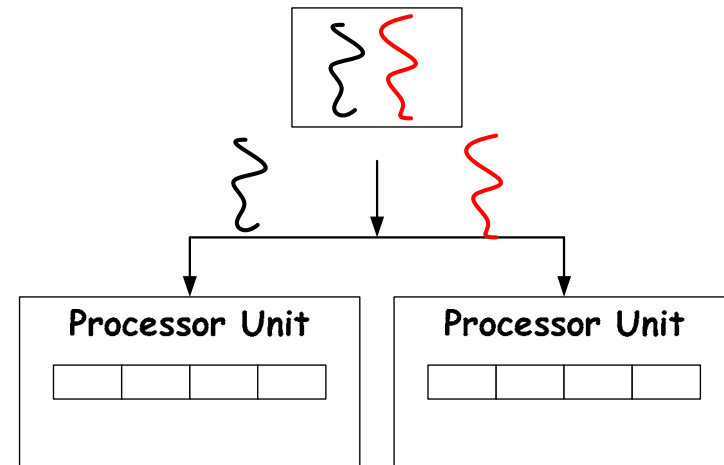
TLP Exploiting at MAJC 5200

- Vertical multithreading
- CMP
- Speculative multithreading



Vertical multithreading

multithread share the resources in a processor unit on a cycle-by-cycle basic



CMP

threads are scheduled to different processor units

- Intel Itanium Line Roadmap
- IA-64 Architecture
- Itanium Processor Microarchitecture
- Case Study: MAJC 5200 Processor