intel

SuperScalar Architecture of the P5 Intel's Next Generation Microprocessor

Donald Alpert

Intel Corporation

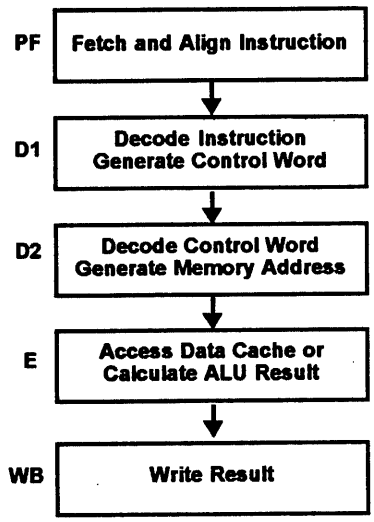
intel

Outline

- Integer Pipeline
- Superscalar Execution
- Branch Prediction
- Dual-Access Data Cache
- Compiler Optimizations

intel

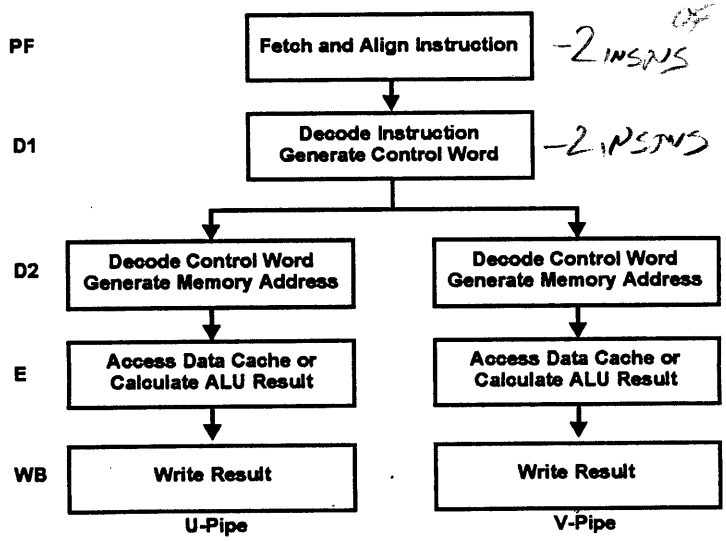
Integer Pipeline



Hot Chips IV

intel

Superscalar Execution



Hot Chips IV

intel

Instruction Issue Algorithm

Decode Two Consecutive Instructions: I1 and I2
If the Following Are All True
- I1 Is a "Simple" Instruction
- I2 Is a "Simple" Instruction
- I1 Is Not a JUMP Instruction
- Destination of I1 ≠ Source of I2
- Destination of I1 ≠ Destination of I2

Then Issue I1 to U-Pipe and I2 to V-Pipe
Else Issue I1 to U-Pipe

90% OF DYNAMIC FREQUENCY

"Simple" Instructions Are Generally ALU or MOV Operations, Including Reg-Reg, Imm-Reg, Mem-Reg, and Reg-Mem Formats, and JUMPS

intel DHRYSTONE BENCHMARK >100 MIPS @ 66MHz

Example

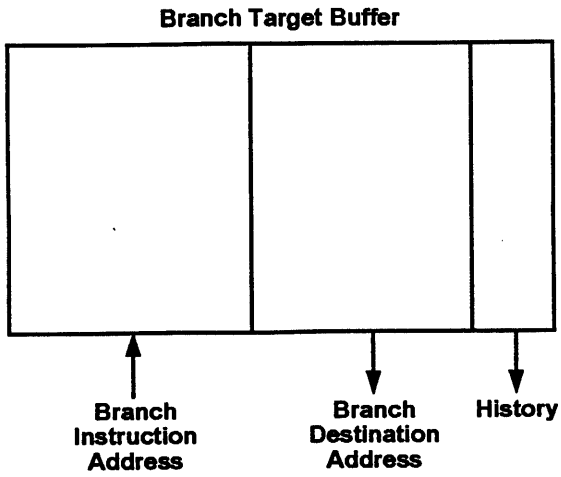
WHAT SHOW PAIR MIPS

	U-Pipe		V-Pipe
Proc2:			
pushl	%ebx	movl	%eax, %ecx
movl	(%ecx), %edx		
addl	$10, %edx	movb	Char1Glob, %ah
cmpb	$65, %ah	jne	.B4_4
decl	%edx	movl	IntGlob, %eax
movl	%edx, %ebx	subl	%eax, %edx
movl	%edx, (%ecx)	movl	%ebx, %edx
.B4_4:			
popl	%ebx		
ret			

↑ THIS BELOW WHAT'S TYPICAL 30%-50% RUN SUPER SCALAR

intel

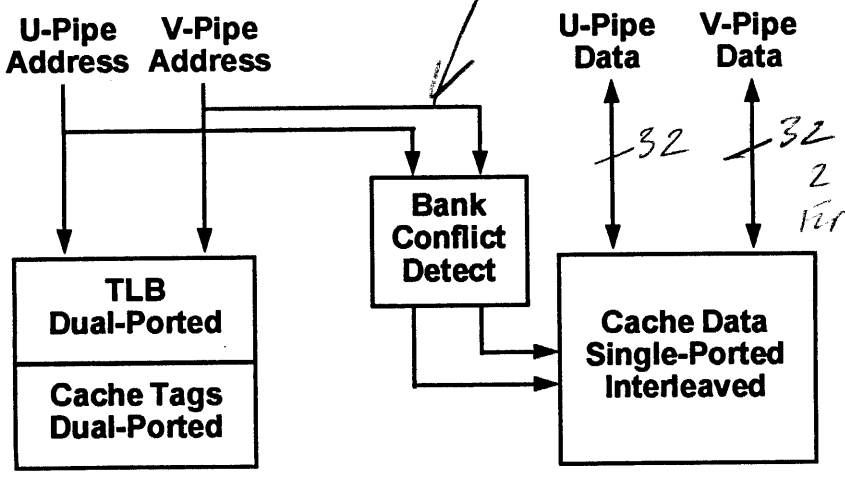
Branch Prediction



Correctly Predicted Branches Execute with No Delays

intel

Dual-Access Data Cache



intel

Compiler Optimization

- Instruction Selection
  - Use Simple Formats for Efficient Decoding
- Instruction Scheduling
  - Minimize Address Generation Interlocks
  - Maximize Parallel Execution
- Register Allocation
  - Schedule and Allocate Together to Make Best Use of Small Register Set

0.8µm Technology

intel

Summary

- Superscalar Microarchitecture
  - Dual Integer Pipelines
  - Branch Target Buffer
  - Dual-Access Data Cache
- Fully Compatible with Intel486™ CPU

intel