

A VLIW Processor for Multimedia Applications

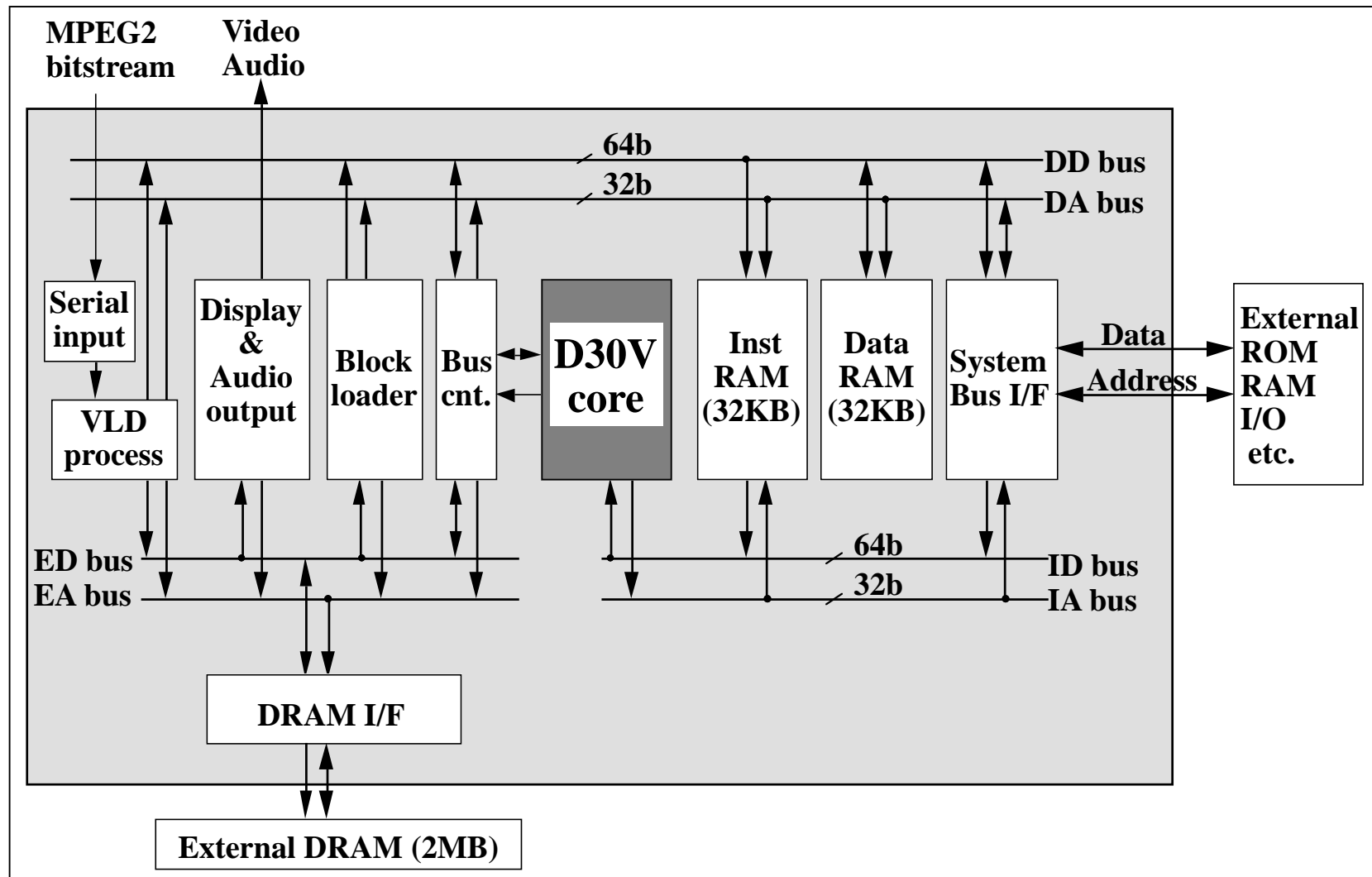
E. Holmann T. Yoshida A. Yamada Y. Shimazu

Mitsubishi Electric Corporation, System LSI Laboratory
4-1 Mizuhara, Itami, Hyogo 664, Japan

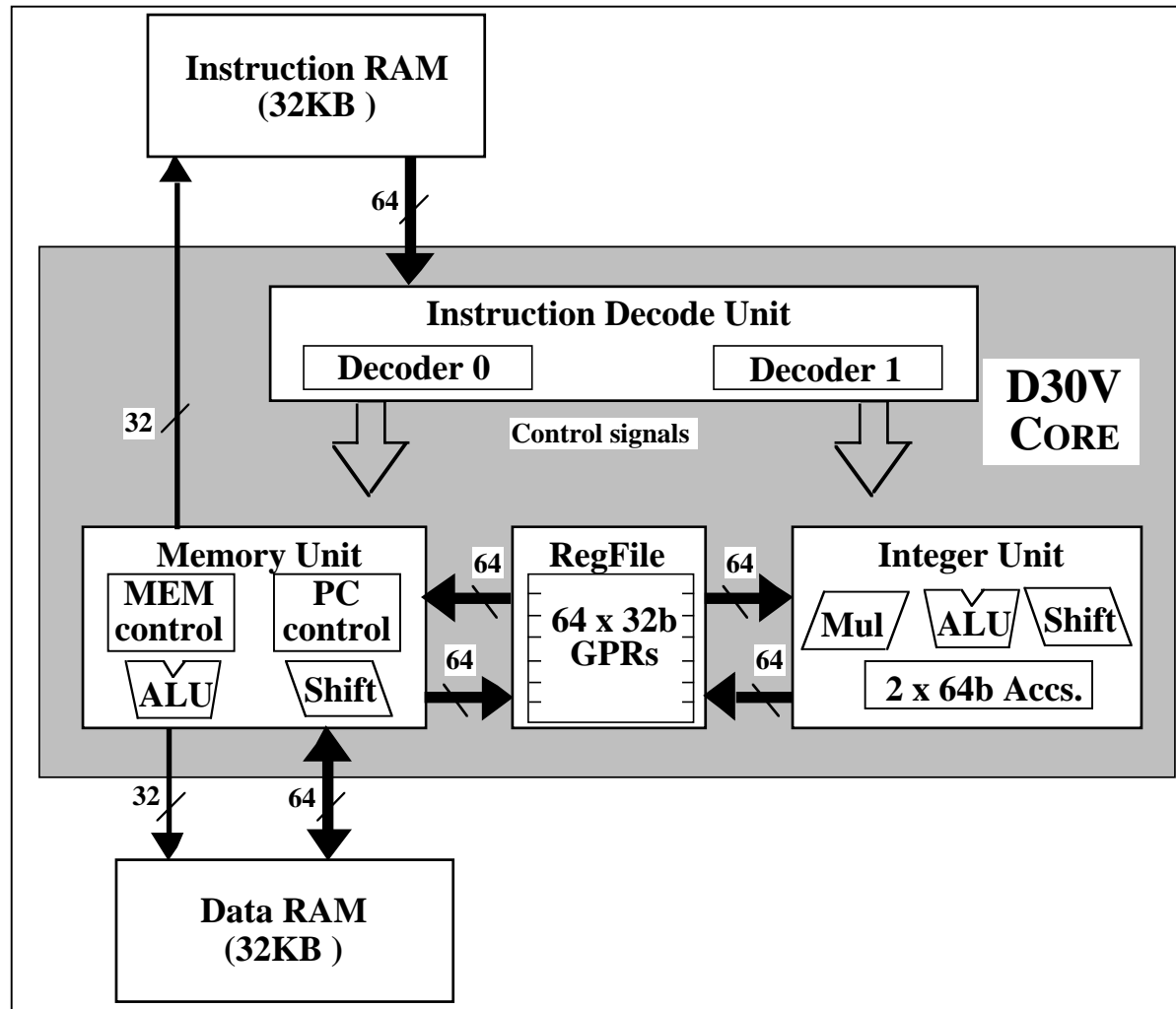
Outline

- Objective
- System Architecture
- Performance
- Conclusions

System Architecture



Processor Core Diagram



Instruction Formats

- Two types of instructions
 - Two short RISC sub-instructions (28 bits each)

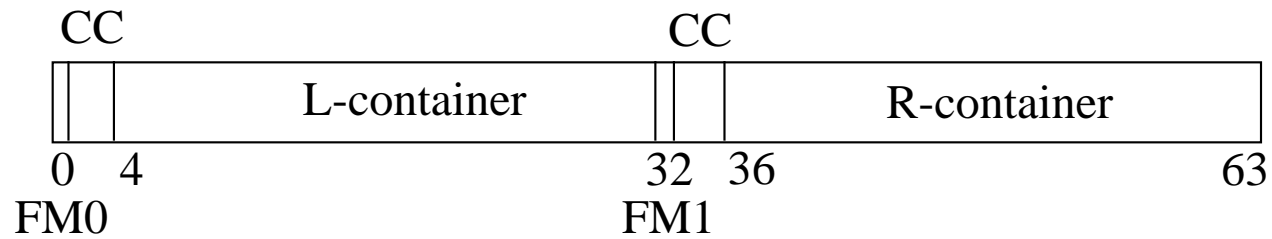
Short sub-instruction L

Short sub-instruction R

- One long RISC sub-instruction (54 bits)

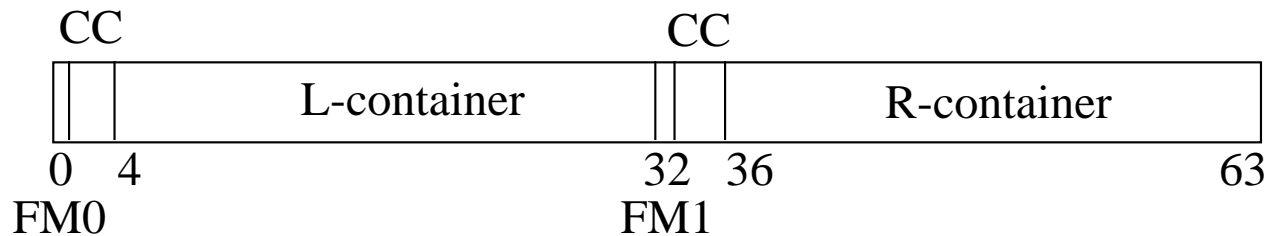
Long sub-instruction

Instruction Issuing



FM		Issue format
00	<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 5px; width: 200px; text-align: center;">Short sub-instruction L</div> <div style="border: 1px solid black; padding: 5px; width: 200px; text-align: center;">Short sub-instruction R</div> </div>	parallel
01	<div style="border: 1px solid black; padding: 5px; width: 200px; margin: 0 auto; text-align: center;">Short sub-instruction L</div> <div style="border: 1px solid black; padding: 5px; width: 200px; margin: 5px auto; text-align: center;">Short sub-instruction R</div>	serial
10	<div style="border: 1px solid black; padding: 5px; width: 200px; margin: 0 auto; text-align: center;">Short sub-instruction R</div> <div style="border: 1px solid black; padding: 5px; width: 200px; margin: 5px auto; text-align: center;">Short sub-instruction L</div>	serial
11	<div style="border: 1px solid black; padding: 5px; width: 400px; margin: 0 auto; text-align: center;">Long sub-instruction</div>	long inst

Speculative Execution



- Every sub-instruction is speculatively executed
- 3 bits define condition for execution
- Conditions are based on status of user flags
- PSW has 8 user flags
- 2 user flags used for speculative execution

ALU Special Operations

➔ Added video operations

- Variable length saturation instruction:
 - SAT, SATZ, SATHL, SATHH
 - » SAT ra, rb, 24 -> ra = saturate (rb, 24)
 - » SATHH ra, rb, 12 -> raH = saturate (rb, 12)
- Flexible join instruction:
 - JOINLL, JOINLH, JOINHL, JOINHH
 - » JOINLH ra, rb, rc -> ra = rbL || rcH
- Add sign instruction
 - ADDS
 - » ADDS ra, rb, rc -> ra = rb + sign(rc)

ALU Special Operations (cont.)

➔ Added sub-word operations

- ALU operations on dual half-word data:
 - ADD2H, SUB2H, ADDS2H, AVG2H, SAT2H, SATZ2H
 - MUL2H, MULX2H
 - » ADD2H ra, rb, rc -> $raH = rbH + rcH$
 $raL = rbL + rcL$
- Shifter operations on dual half-word data:
 - SRA2H, SRL2H, ROT2H
 - » SRA2H ra, rb, 3 -> $raH = rbH \gg 3$
 $raL = rbL \gg 3$

ALU Special Operations (cont.)

➔ **Added single half-word operations**

- ALU operations on single half-word operands
 - ADDHppp, SUBHppp, JOINpp, MULHXpp, SATHp
 - » ADDHLHH ra, rb, rc -> $raL = rbH + rcH$
 - » SUBHHLH ra, rb, rc -> $raH = rbL - rcH$
 - » JOINLH ra, rb, rc -> $ra = rbL || rcH$

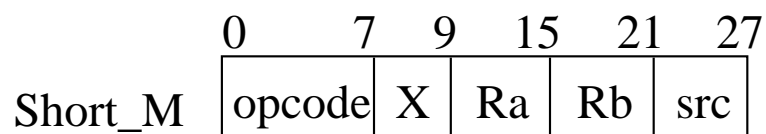
Memory Unit Special Features

- Flexible operand types:
 - byte (signed, unsigned)
 - half-word (signed, unsigned)
 - word
 - double word
- Multiple operand accessing:
 - four byte data load with packing
 - four byte data store with unpacking
 - two half-word data load with packing
 - two half-word data store with unpacking
- Post-increment/decrement register indexed
- Modulo addressing

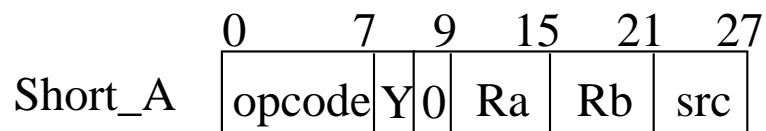
Branch Unit Special Features

- Destination address calculated in second pipe stage
- Variable number of delay slots
- Block repeat with zero delay penalty
- Additional conditional branches
 - Test zero and branch instruction
 - Test not-zero and branch instruction

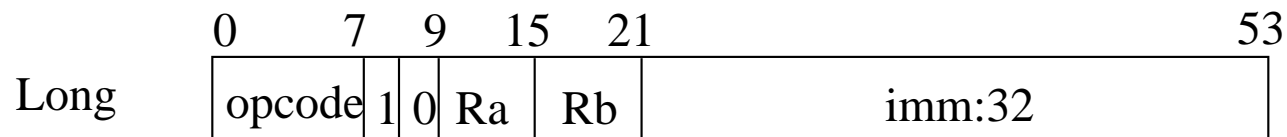
Instruction Examples



LDBU R7, @(R6, 20)
 LDW R4, @(R5, R7)
 LD2W R8, @(R7+, R22)

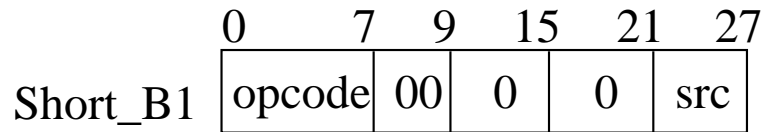


ADD R7, R6, R8
 SUB R10, R6, 20
 ADD2H R4, R5, R7

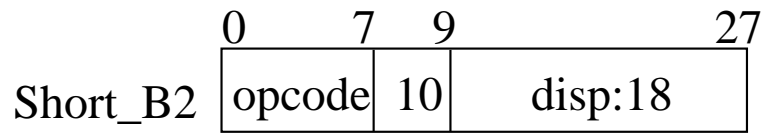


LD2H R7, @(R6, 0x00001000)
 AVG2H R8, R9, 0x00010001

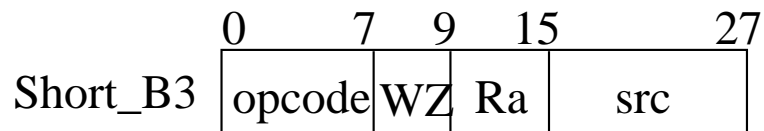
Instruction Examples (Branches)



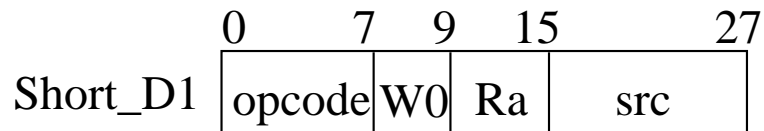
BRA R3
JMP R4



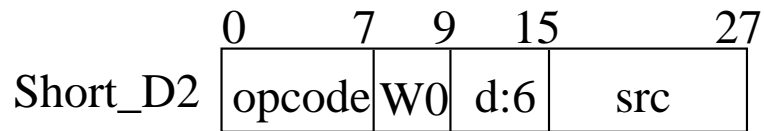
BRA 0x1234
JMP 0x10000



BSRTZR R2, R5
JMPTNZ R52, 0x200



DBRA R3, R17
DBRA R3, 0x39A



DJMP 3, R50
DBSR 41, 0x300

Pipeline Specification

ALU	IF	D	EX	WB
-----	----	---	----	----

IF : Instruction Fetch
D : Decode
EX : Execute
WB : Write Back

LD/ST	IF	DA	M	WB
-------	----	----	---	----

IF : Instruction Fetch
DA : Decode & Address
M : Memory
WB : Write Back

BRA	IF	DA	EX	WB
-----	----	----	----	----

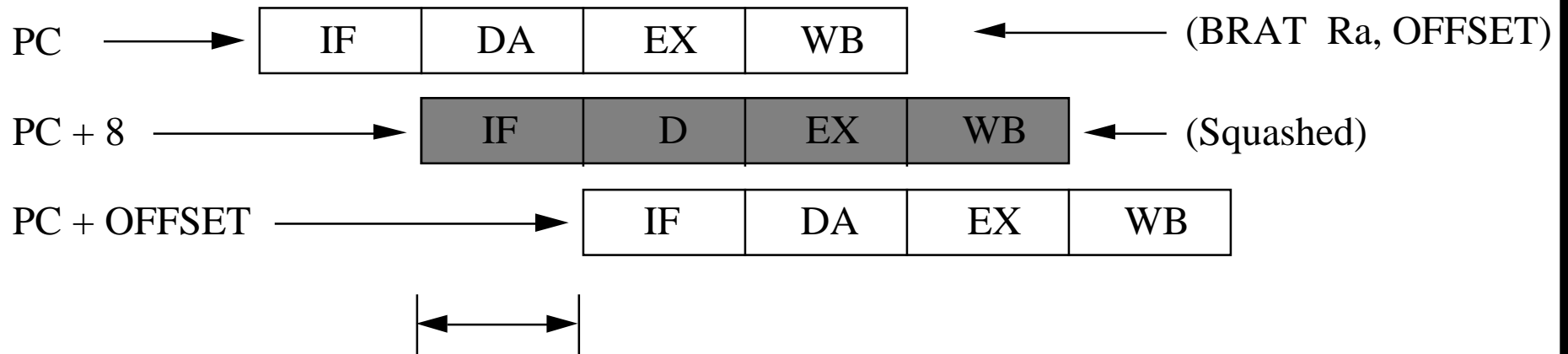
IF : Instruction Fetch
DA : Decode & Address
EX : Execute (delayed branch)
WB : Write Back

MUL16	IF	D	EX	WB
-------	----	---	----	----

IF : Instruction Fetch
D : Decode
EX : Execute
WB : Write Back

Conditional Branch Instructions

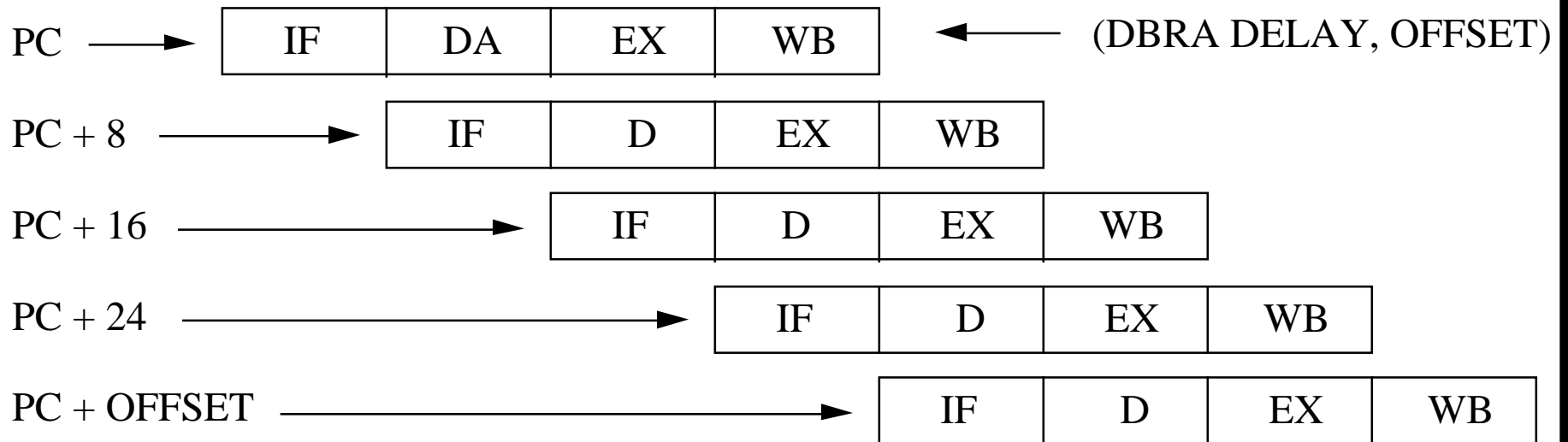
Instructions: BRAT, BSRT, JMPT, JSRT



- Decode Instruction
- Calculate newPC
- Speculative Execution (CC bits and user flags)
- Test register for zero/not zero (conditional execution)

Delayed Branch Instructions

Instructions: DBRA, DBSR, DJMP, DJSR



Decode Instruction
 Calculate PC + offset
 Speculative Execution

Calculate PC + delay

Processor Parameters and Figures of Merit

Clock Frequency	250 MHz
Parallelism	2 way VLIW, 2 way SIMD
Peak Performance	1000 MIPS
Register File	64 x 32bits
RAM	32KB DRAM, 32KB IRAM
8x8 IDCT	< 2 μ seconds
256 point complex IFFT	~ 40 μ seconds
MPEG-2 macroblock	< 800 cycles (real time)

Conclusions

- High performance dual-issue RISC system
 - zero delay branches
 - zero delay repeat loops
 - speculative execution
- Multimedia Processor
 - sub-word operations
 - half-word operations
 - special video operations
- Single chip system for DSP applications
 - D30V serves functions of DSP and MCU chip

Application areas for D30V

- 2D/3D graphics
- AC-3 decode
- modem V.34 (28.8kbps)
- H.263 codec
- MPEG-1 decode
- MPEG-2 decode