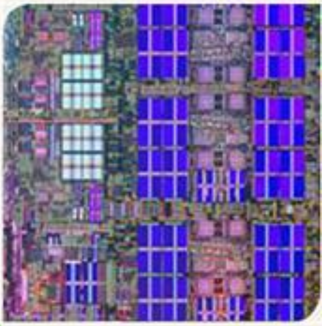
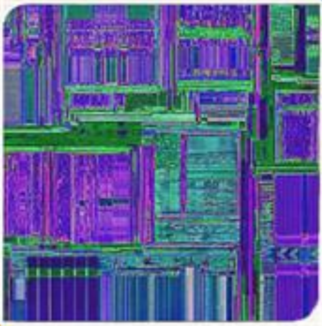




CS4803DGC Design Game Consoles

Spring 2010

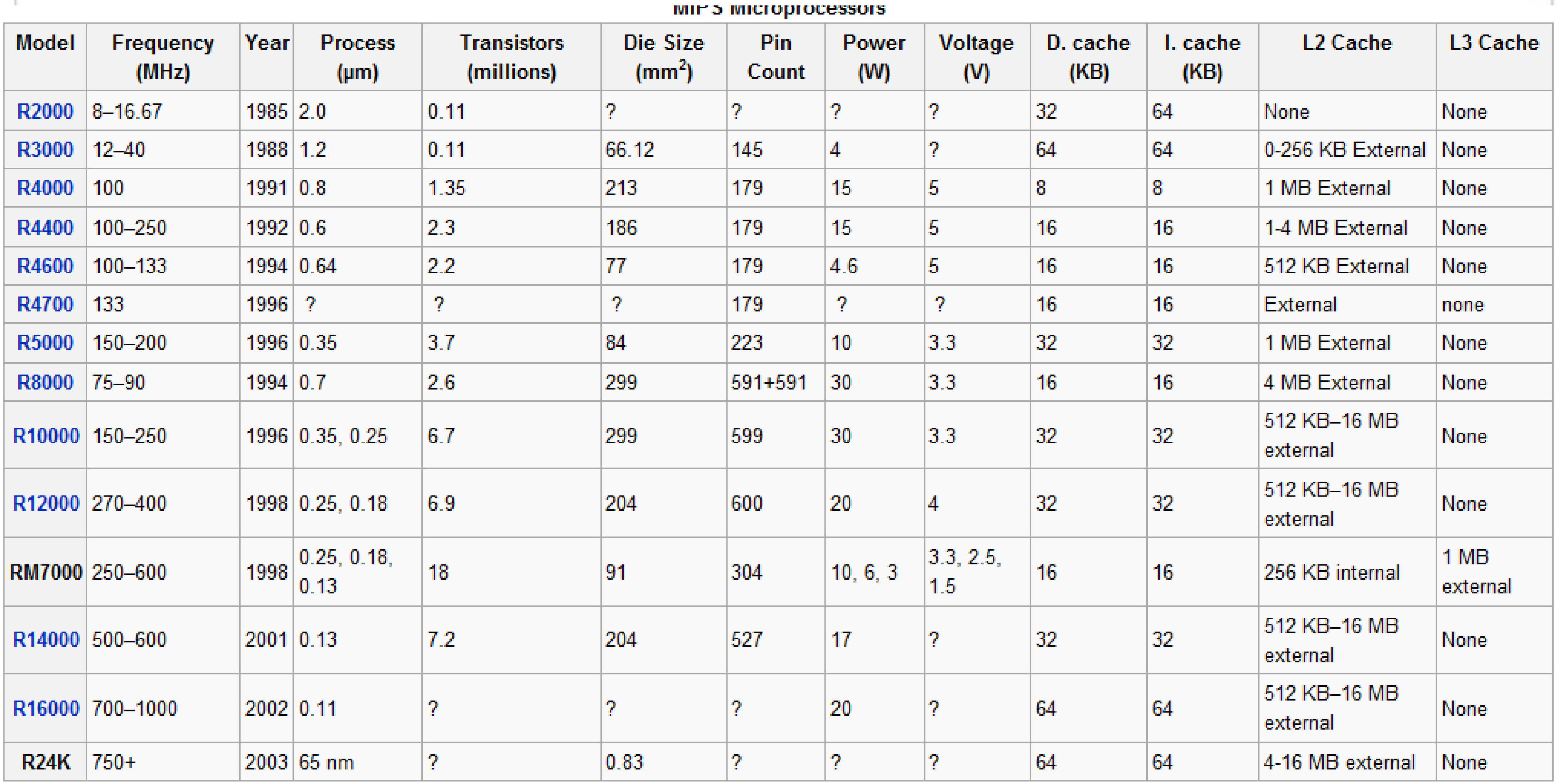
Prof. Hyesoon Kim

Georgia Tech College of Computing

MIPS

- MIPS (Microprocessor without interlocked pipeline stages)
- MIPS Computer Systems Inc.
- MIPS architecture usages
- 1990’s
  - – R2000, R3000, R4000, Motorola 68000 family
- Playstation, Playstation 2, Sony PSP handheld, Nintendo 64 console
- Android

MIPS Architecture Diagrams



MIPS Microprocessors

Model	Frequency (MHz)	Year	Process (µm)	Transistors (millions)	Die Size (mm^2)	Pin Count	Power (W)	Voltage (V)	D. cache (KB)	I. cache (KB)	L2 Cache	L3 Cache
R2000	8–16.67	1985	2.0	0.11	?	?	?	?	32	64	None	None
R3000	12–40	1988	1.2	0.11	66.12	145	4	?	64	64	0-256 KB External	None
R4000	100	1991	0.8	1.35	213	179	15	5	8	8	1 MB External	None
R4400	100–250	1992	0.6	2.3	186	179	15	5	16	16	1-4 MB External	None
R4600	100–133	1994	0.64	2.2	77	179	4.6	5	16	16	512 KB External	None
R4700	133	1996	?	?	?	179	?	?	16	16	External	none
R5000	150–200	1996	0.35	3.7	84	223	10	3.3	32	32	1 MB External	None
R8000	75–90	1994	0.7	2.6	299	591+591	30	3.3	16	16	4 MB External	None
R10000	150–250	1996	0.35, 0.25	6.7	299	599	30	3.3	32	32	512 KB–16 MB external	None
R12000	270–400	1998	0.25, 0.18	6.9	204	600	20	4	32	32	512 KB–16 MB external	None
RM7000	250–600	1998	0.25, 0.18, 0.13	18	91	304	10, 6, 3	3.3, 2.5, 1.5	16	16	256 KB internal	1 MB external
R14000	500–600	2001	0.13	7.2	204	527	17	?	32	32	512 KB–16 MB external	None
R16000	700–1000	2002	0.11	?	?	?	20	?	64	64	512 KB–16 MB external	None
R24K	750+	2003	65 nm	?	0.83	?	?	?	64	64	4-16 MB external	None

http://en.wikipedia.org/wiki/MIPS_architecture

MIPS in Game Consoles: PSP Spec

- MIPS R4000 CPU core
- Floating point and vector floating point co-processors
- 3D-CG extended instruction sets
- Graphics
  - 3D curved surface and other 3D functionality
  - Hardware clipping, compressed texture handling

- R4300 (embedded version) – Nintendo-64

http://www.digitaltrends.com/gaming/sony-announces-playstation-portable-specs/

MIPS ISAs

- Started from 32-bit
- Later 64-bit
- 16-bit compression version (similar to ARM thumb)
- SIMD additions-64 bit floating points

http://www.spiritus-temporis.com/mips-architecture/

Conditional Move Instructions

- Conditionally move one CPU general register to another
- Limited form of predicated execution.
    - Difference between fully predicated execution and conditional move?

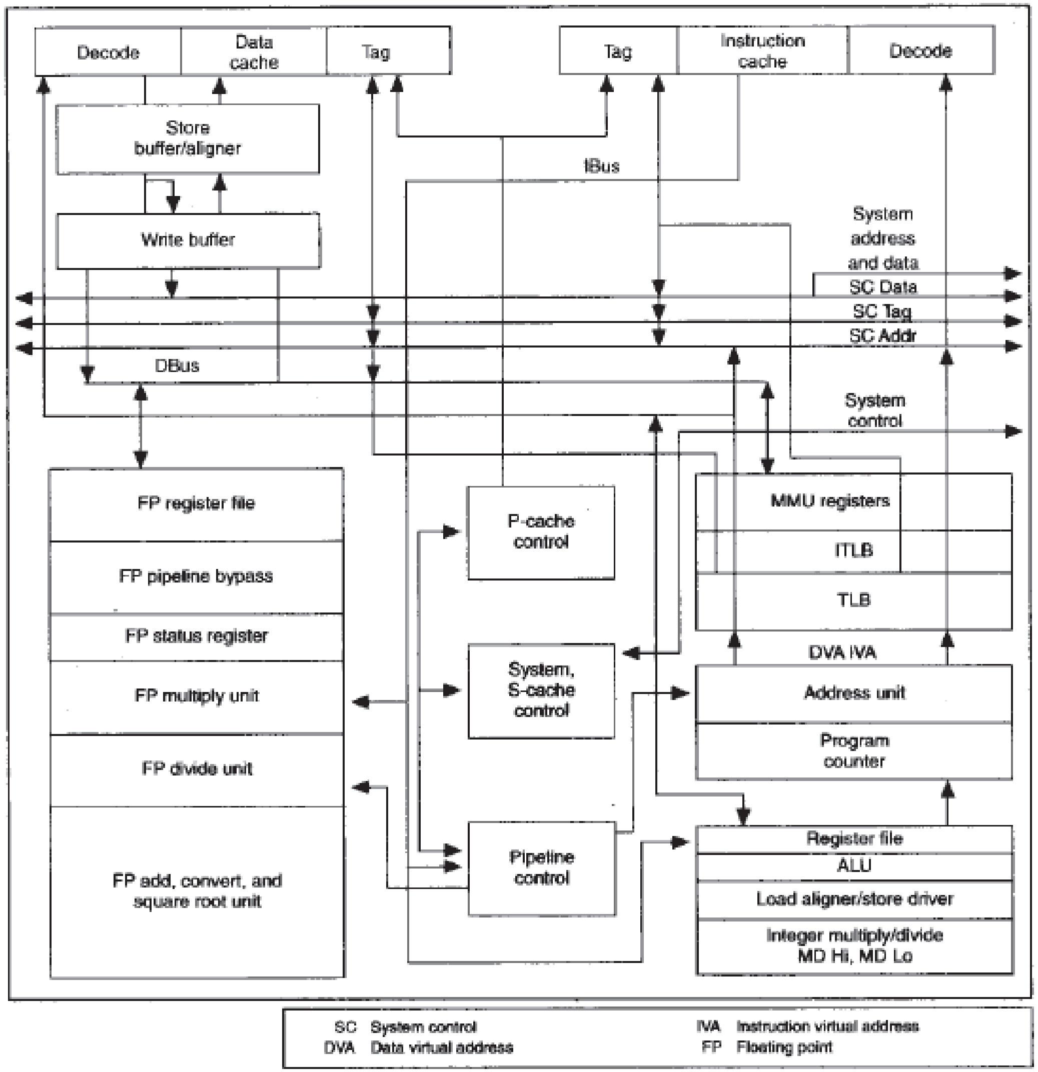
MIPS ISA

- **32-bit fixed format inst** (3 formats)
- **31 32-bit GPR** (R0 contains zero) and 32 FP registers (and HI LO)
  - partitioned by software convention
- **3-address, reg-reg arithmetic instr.**
- **Single address mode for load/store:** base+displacement
- **Simple branch conditions**
  - compare one register against zero or two registers for =,≠
  - no condition codes for integer operations

MIPS R4000

The Mips R4000 Processor,
Mirapuri, S.; Woodacre, M.; Vasseghi, N.;
Micro, IEEE Volume: 12 , Issue: 2
Publication Year: 1992 , Page(s): 10 - 22

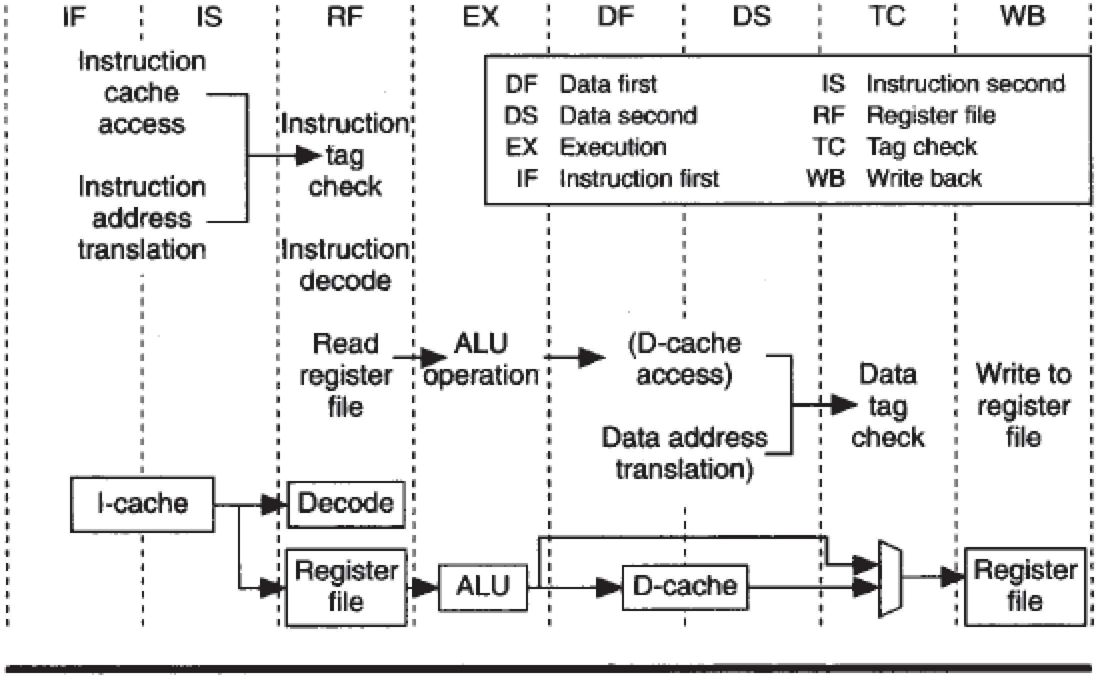
Pipeline



P-cache: Primary cache
S-cache: Secondary cache

The Mips R4000 Processor, Mirapuri, Woodacre,Vasseghi, N., ‘92

Pipeline

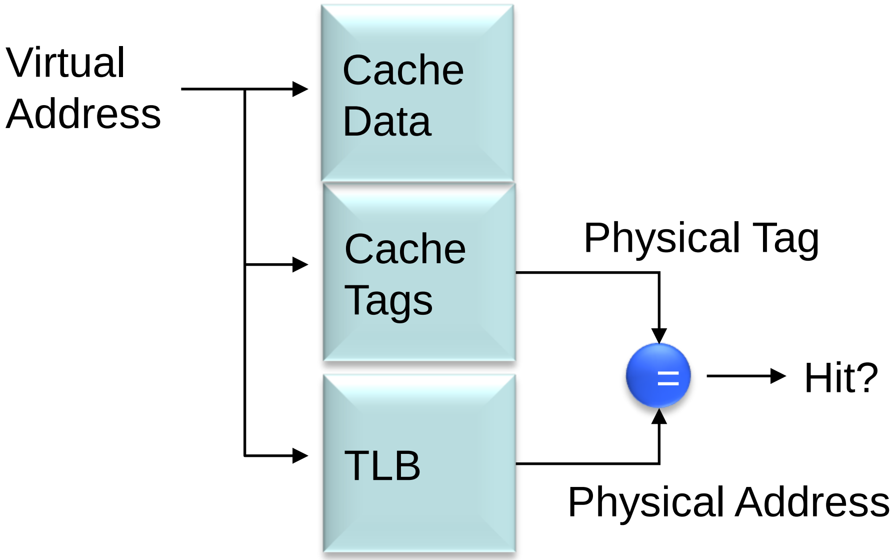


DF	Data first	IS	Instruction second
DS	Data second	RF	Register file
EX	Execution	TC	Tag check
IF	Instruction first	WB	Write back

Figure 2. R4000 pipeline activities.

TAG Check

- Q: Tag check stage, why is it at the end of load access?
- A: virtual indexed physically tagged (VIPT)



Load Delay Slots

R2000 load has a delay slot

LW ra ---

Addi ra rb rc See old Ra value (before load)

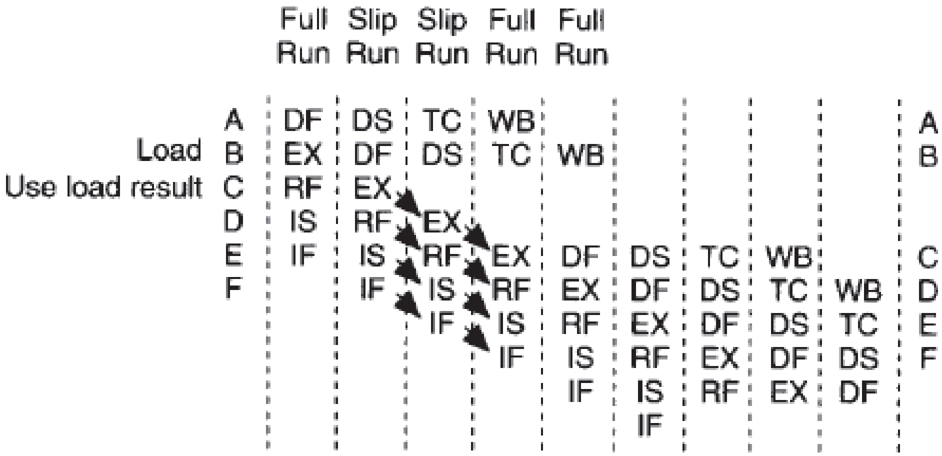
Addi ra rb rc

Good idea? Bad Idea?

R4000 does not have load delay slots.

Handling Load: Slip

- 2-cycle delay loads
- Data is not available until the end of DS
- Only DF/DS/TC/WB stages make a progress for load instructions (IS/RF/EX pipeline stages stall)



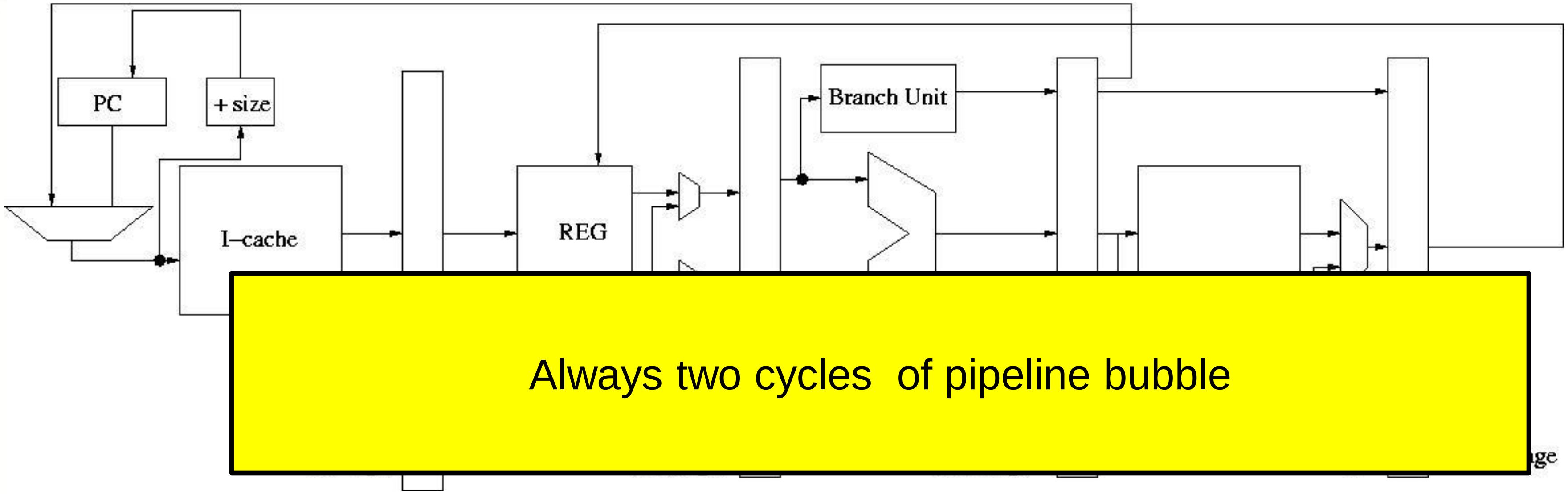
		Full Run	Slip Run	Slip Run	Full Run	Full Run					
	A	DF	DS	TC	WB						A
Load	B	EX	DF	DS	TC	WB					B
Use load result	C	RF	EX								
	D	IS	RF	EX							
	E	IF	IS	RF	EX	DF	DS	TC	WB		C
	F		IF	IS	RF	EX	DF	DS	TC	WB	D
				IF	IS	RF	EX	DF	DS	TC	E
					IF	IS	RF	EX	DF	DS	F
						IF	IS	RF	EX	DF	
							IF				

Figure 4. Load interlock/slip cycle.

Memory Hierarchy

- 2-level cache hierarchy
- Different line sizes
  - Pros? cons?
- Inclusive cache
- Primary cache: initial design 8BKB → 32KB
  - Direct-mapped, VIPT
  - 16 or 32B software programmable line size
- Secondary cache
  - 128-bit, up to 4MB

Handling Branches



Always two cycles of pipeline bubble

cycle	PC (latch)	FE	ID	EX	MEM	WB
1	0x800	br				
2	0x804	add	br			
3	0x804	add	☁	br		
4	0x900	sub	☁	☁	br	
5	0x904	add	sub	☁	☁	br
6	0x908	mul	add	sub	☁	☁

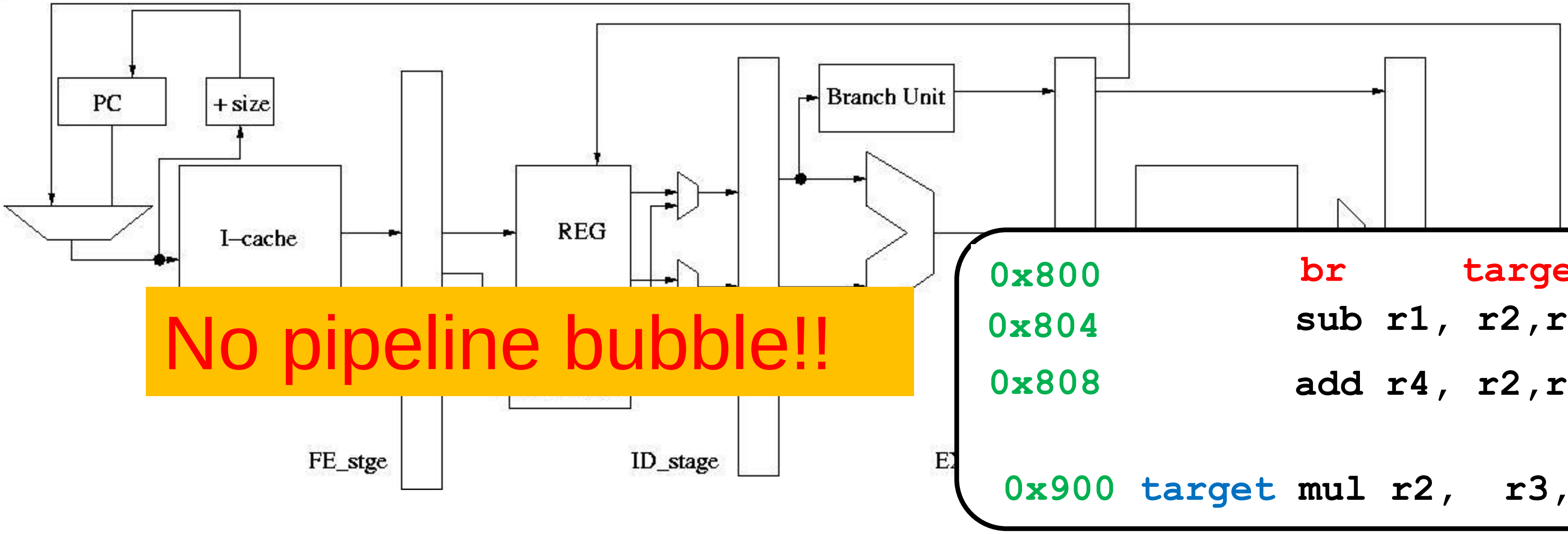
What if we

```
0x800         sub r1, r2,r3
0x804         add r4, r2,r3
0x808         br      target
0x80b
0x810
0x900  target mul r2,  r3,r4
```

```
0x900  target mul r2,  r3,r4
```

Change the rule!
Always execute the next two instructions after a branch

Rule: Always execute the next two instructions after a branch



No pipeline bubble!!

```
0x800            br          target
0x804            sub r1, r2,r3
0x808            add r4, r2,r3

0x900 target mul r2,  r3,r4
```

cycle	Fetch addr	FE	ID	EX	MEM	WB
1	0x800	br				
2	0x804	sub	br			
3	0x808	add	sub	br		
4	0x900	mul	add	sub	br	
5	0x904	div	mul	add	sub	br
6	0x908	add	div	mul	add	sub
7	0x90b	sub	add	div	mul	add

Delayed branch

- N-cycle delay slot
- The compiler fills out useful instructions inside the delay slot

Still Cancel or nullifying instructions

- Different options:
  - Fill the slot from before the branch instruction
    - Restriction: branch must not depend on result of the filled instruction
  - Fill the slot from the target of the branch instruction
    - Restriction: should be OK to execute instruction even if not taken
  - Fill the slot from fall through of the branch
    - Restriction: should be OK to execute instruction even if taken

Branch and Branch Likely

- Branch:
  - Execute the instructions in the delay slot
- Branch likely
  - Do not execute instructions in the delay slot if the branch is not taken
- No not use branch likely!
  - It won’t be supported in the future

Remark

- Many DSP architecture, older RISC, MIPS, PA-RISC, SPARC.
- Delayed branches are architecturally ~~invisible~~ visible
  - Advantage:
    - better performance
  - Disadvantage:
    - what if implementation changes?
    - Deeper pipeline-> more branch delays?
- Interrupt/exceptions?
  - Where to go back?
- Combining with a branch predictor?

MIPS R10000

MIPS R10000

- Later designs are based on R10K
- Out-of-order super scalar processor
- ROB, 32 in-flight instructions
- 4-instruction wide