CS4803DGC Design and Programming of Game Console

Spring 2011

Prof. Hyesoon Kim

Georgia Tech | College of Computing

MIPS Architecture

MIPS

- MIPS (Microprocessor without interlocked pipeline stages)
- MIPS Computer Systems Inc.
- Developed from Stanford
- MIPS architecture usages
- 1990's
  - R2000, R3000, R4000, Motorola 68000 family
- Playstation, Playstation 2, Sony PSP handheld, Nintendo 64 console
- Android
- Shift to SOC

MIPS Architecture Diagrams

| Model	Frequency (MHz)	Year	Process (μm)	Transistors (millions)	Die Size (mm²)	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

Georgia Tech | College of Computing

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

Georgia Tech | College of Computing

PSP
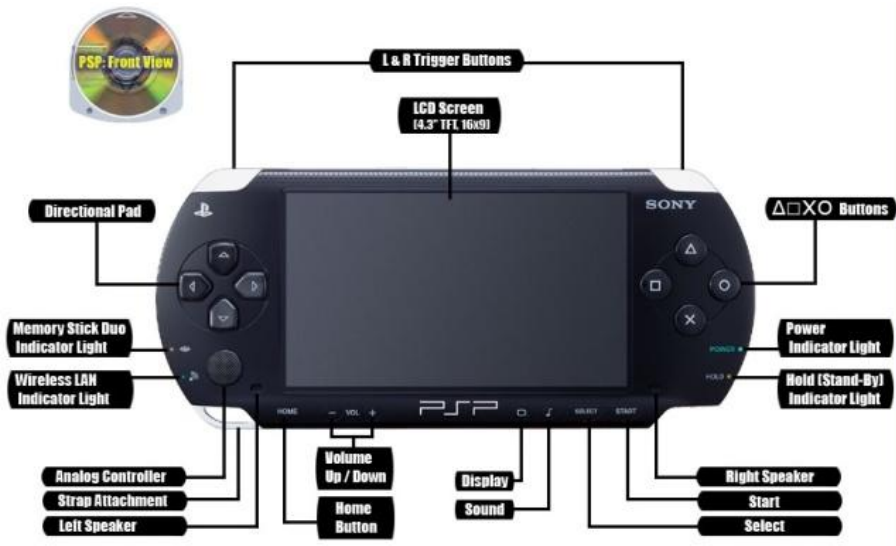
Not Yet out

MIPS in Google TV

- Google TV: an Android-based software service that lets users switch between their TV content and Web applications such as Netflix and Amazon Video on Demand
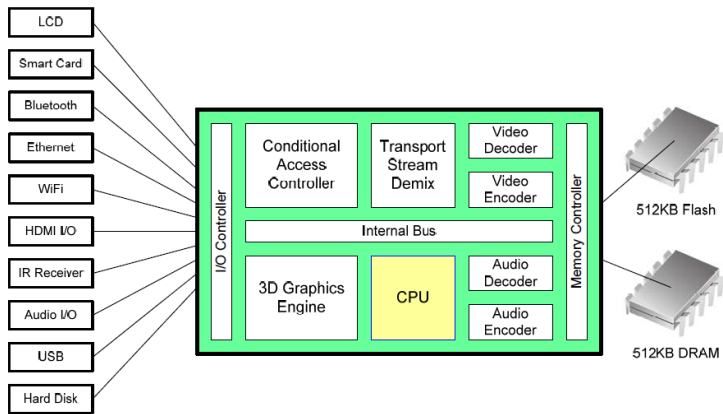
- GoogleTV : search capabilities.
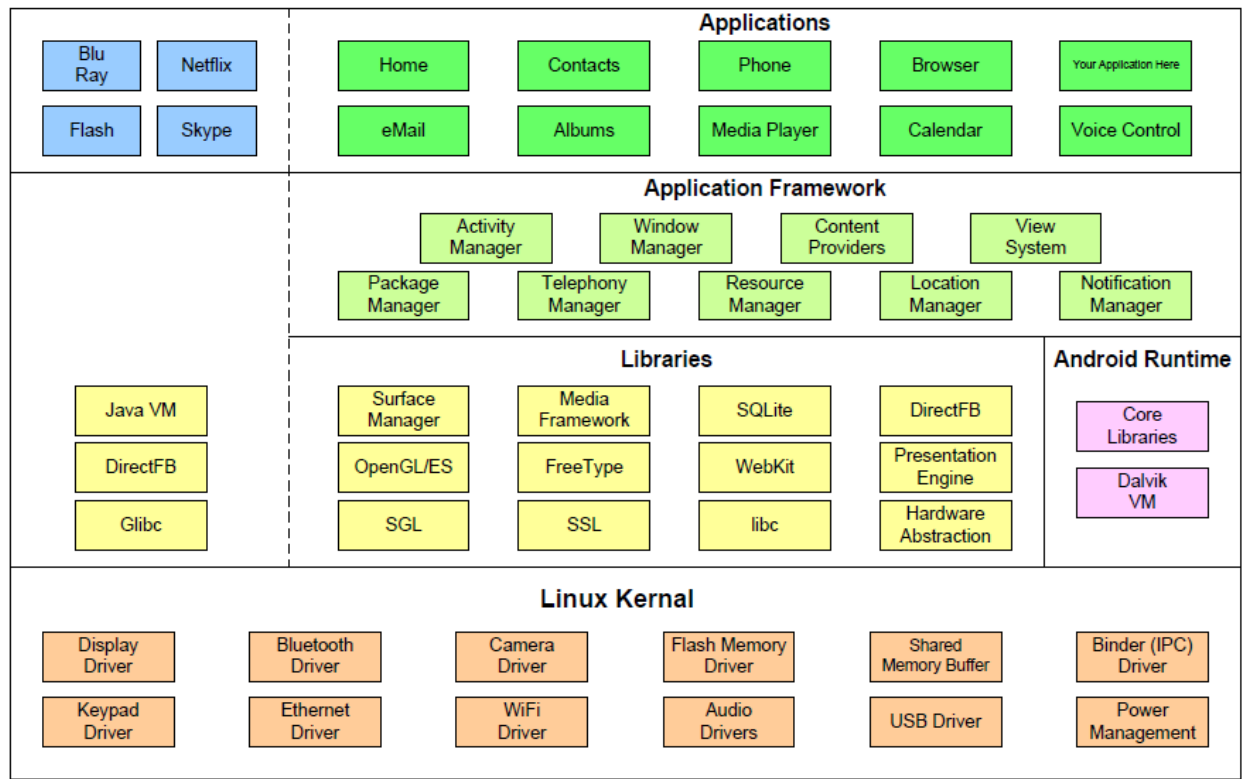
Figure 1 Google TV Hardware Diagram

Figure 3 Google TV Operating Environment

Unpredictable Workloads

- High stream data?

- Internet accesses?

- Multi-threading, SMP design

- High graphics processors

- Several CODEC
  - Hardware vs. Software

- Displaying frame buffer

  e.g) 1080p resolution:

   1920 (H) x 1080 (V)

      color depth: 4 bytes/pixel

      4*1920*1080 ~= 8.3MB

  8.3MB * 60Hz=498MB/sec

CODEC	Description
H.264[3]	Web & Adobe Flash Video - 1080p, 60fps
VP6	Adobe Flash Video - 1080p, 60fps
VP8	HTML5 Video
Sorenson Video	Adobe Flash Video
Ogg Theora	Open source video format used in Google Chrome and YouTube
Ogg Vorbis	Open source audio format used in Google Chrome and YouTube
RealVideo	RealVideo is a proprietary video format developed by RealNetworks
RealAudio	RealAudio is a proprietary audio format developed by RealNetworks
WMV	Windows Media Video format
MOV	Apple QuickTime movie format
MJPEG	Motion JPEG video format
AVI	Audio Video interleave format used by Microsoft
FLV	Adobe – FLV1 video codec, MP3 audio
JPEG	Compression format for still images
PNG	Patent free still image compression format
GIF	Graphics interchange format for still image compression
WMA	Windows Media Audio
MP3/MP2/MPA	MPEG audio formats
MIDI	MIDI music synthesis, requires synthesis engine
WAV	Audio for Windows format
AAC	Advanced Audio Coding
G.729 (optional)	VOIP audio conferencing format

MIPS ISAs

- Started from 32-bit
- Later 64-bit
- microMIPS: 16-bit compression version (similar to ARM thumb)
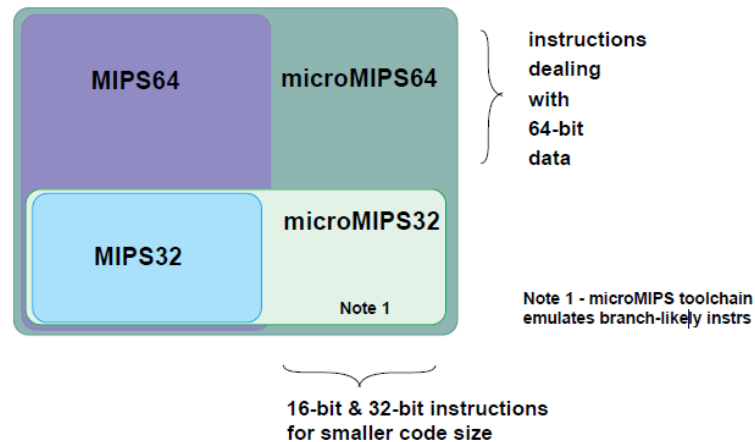
- SIMD additions-64 bit floating points
- User Defined Instructions (UDIs) → coprocessors
- All self-modified code
- Allow unaligned accesses

MIPS64

- 32 64-bit general purpose registers (GPRs)
- A pair of special-purpose registers to hold the results of integer multiply, divide, and multiply-accumulate operations (*HI and LO)*
  - *HI—Multiply and Divide register higher result*
  - *LO—Multiply and Divide register lower result*
- a special-purpose program counter (*PC),*
- A MIPS64 processor always produces a 64-bit result

- 32 floating point registers (FPRs). These registers are 32 bits wide in a 32-bit FPU and 64 bits wide on a 64-bit FPU.
- Five FPU control registers are used to identify and control the FPU.
- Eight floating point condition codes that are part of the *FCSR register*

MIPS 64-bit Cores

	5Kc®	5Kf®	20Kc™
Pipeline Design	MIPS64 6-Stage	MIPS64 6-Stage	MIPS64 7-Stage
Issue Ratetd	Limited Dual Issue	Limited Dual Issue	Full Dual Issue
Synthesizable or Hard	Synthesizable	Synthesizable	Custom Hard Core
Fast Multiplier	yes	yes	yes
Full MMU	yes	yes	yes
Cache Controller	yes	yes	yes
Max Cache Size	64KB	64KB	32KB (fixed)
Date Cache Type	Write Back	Write Back	Write Back
Floating Point Unit	no	yes	yes
MIPS3D ASE	no	no	yes

Georgia Tech | College of Computing

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
  - R0: zero (hardwired value)
- **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 $=, \neq$
  - no condition codes for integer operations

MIPS R4000

Georgia Tech | College of Computing
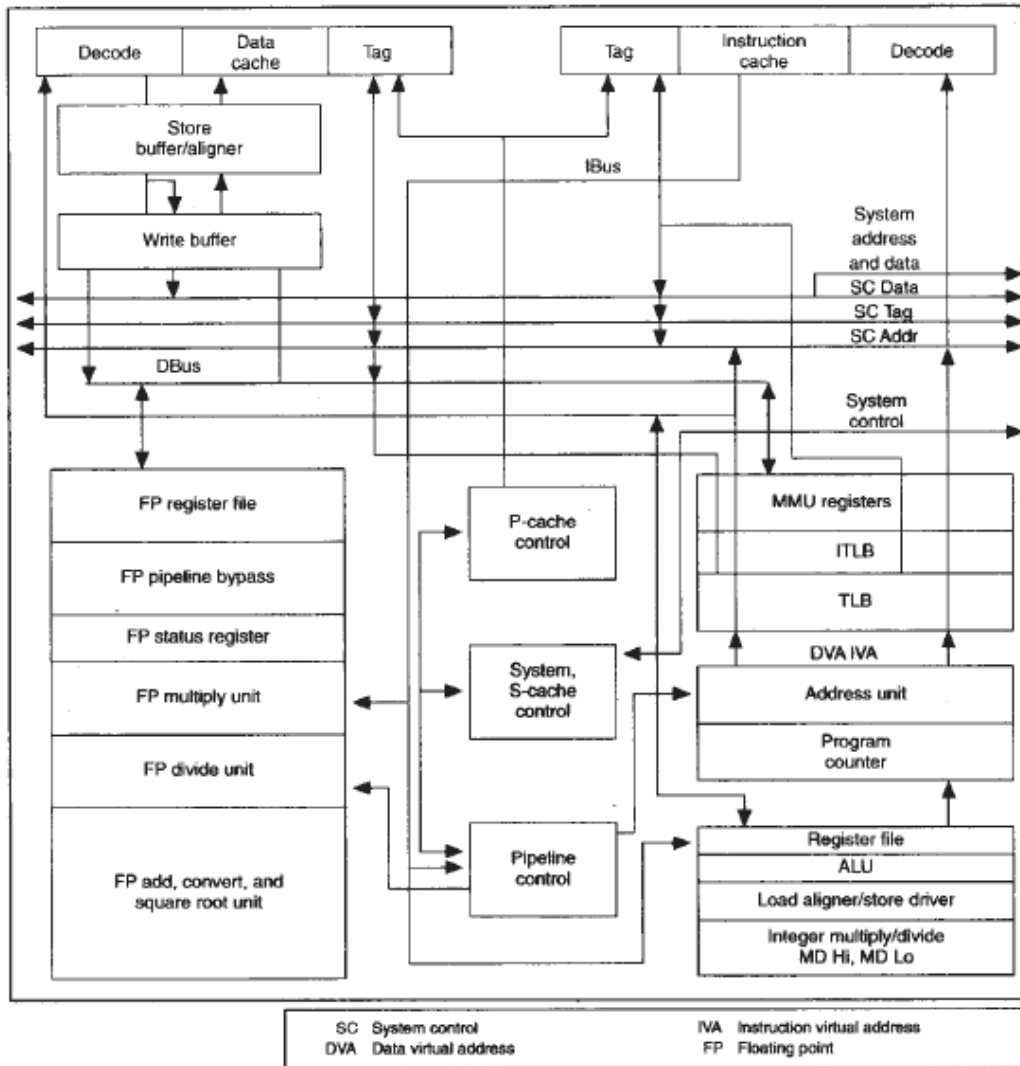
Pipeline

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

Georgia Tech — College of Computing
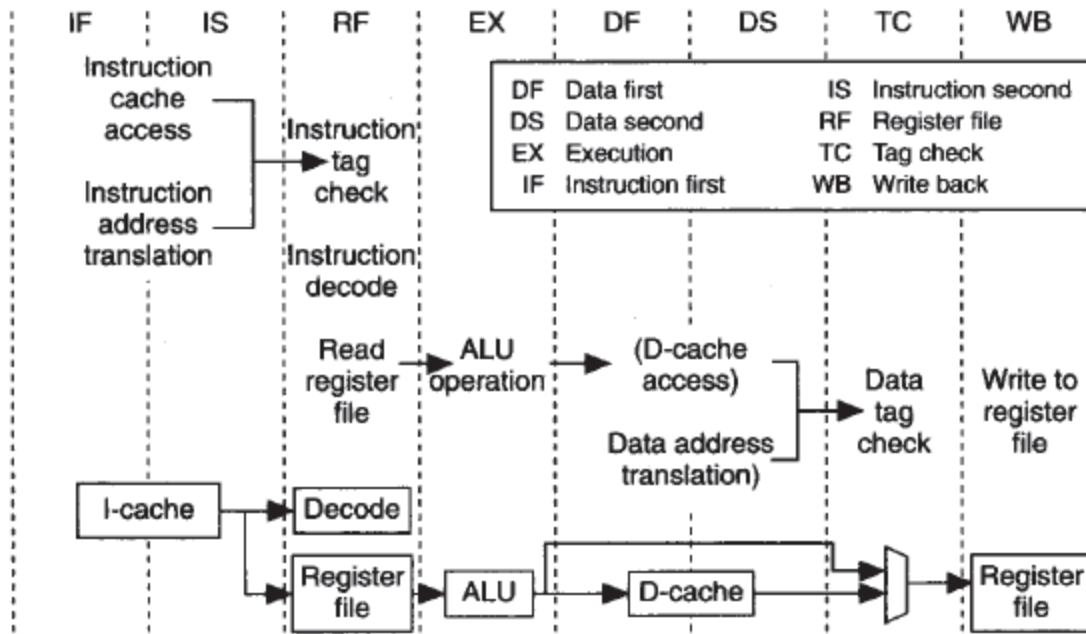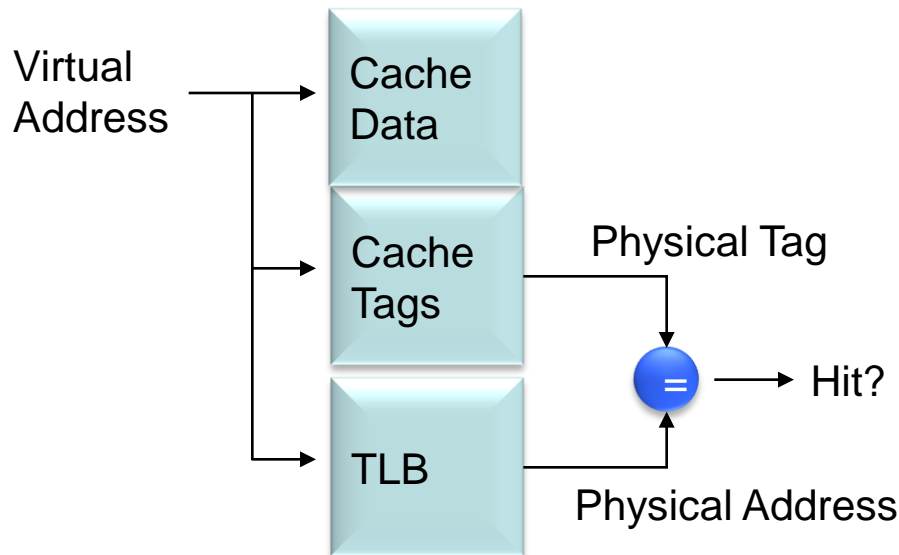
Pipeline

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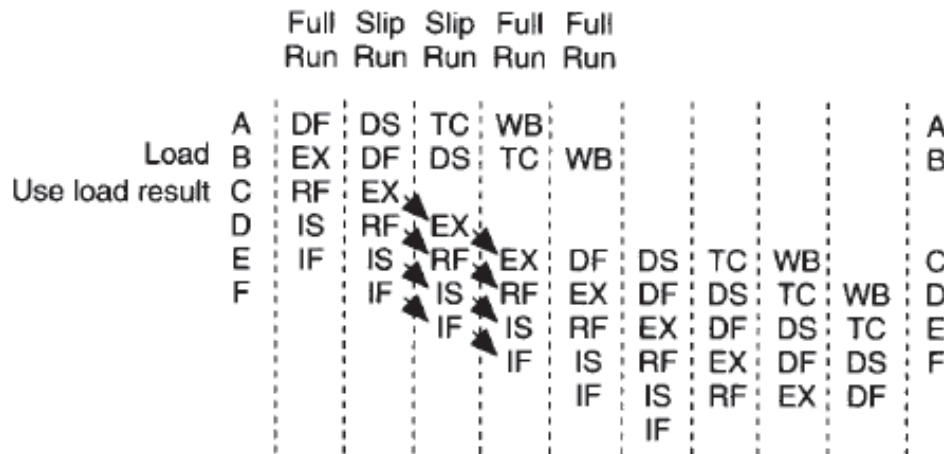
|	Full Run	Slip Run	Slip Run	Full Run	Full Run						
A	DF	DS	TC	WB							A
Load B	EX	DF	DS	TC	WB						B
Use load result C	RF	EX									
D	IS	RF	EX								
E	IF	IS	RF	EX	DF	DS	TC	WB			C
F		IF	IS	RF	EX	DF	DS	TC	WB		D
			IF	IS	RF	EX	DF	DS	TC		E
				IF	IS	RF	EX	DF	DS		F
					IF	IS	RF	EX	DF		
						IF					

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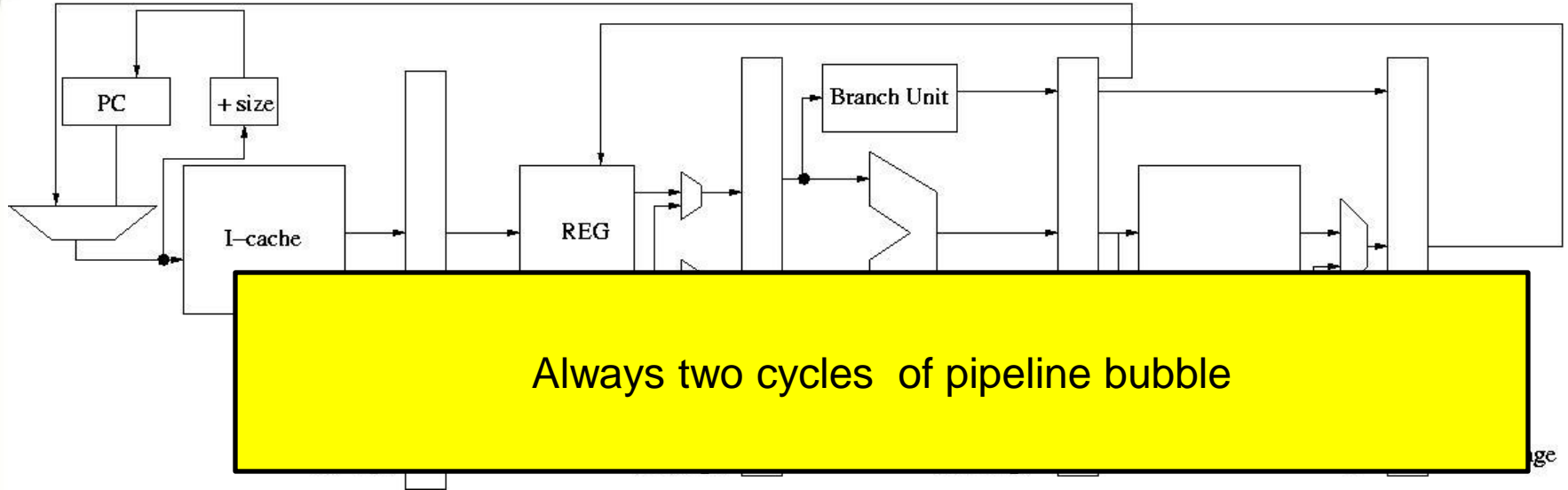
| cycle	PC (latch)	FE	ID	EX	MEM	WB
1	0x800	**br**				
2	0x804	add	**br**			
3	0x804	add	☁	**br**		
4	0x900	sub	☁	☁	**br**	
5	0x904	add	sub	☁	☁	**br**
6	0x908	mul	add	sub	☁	☁

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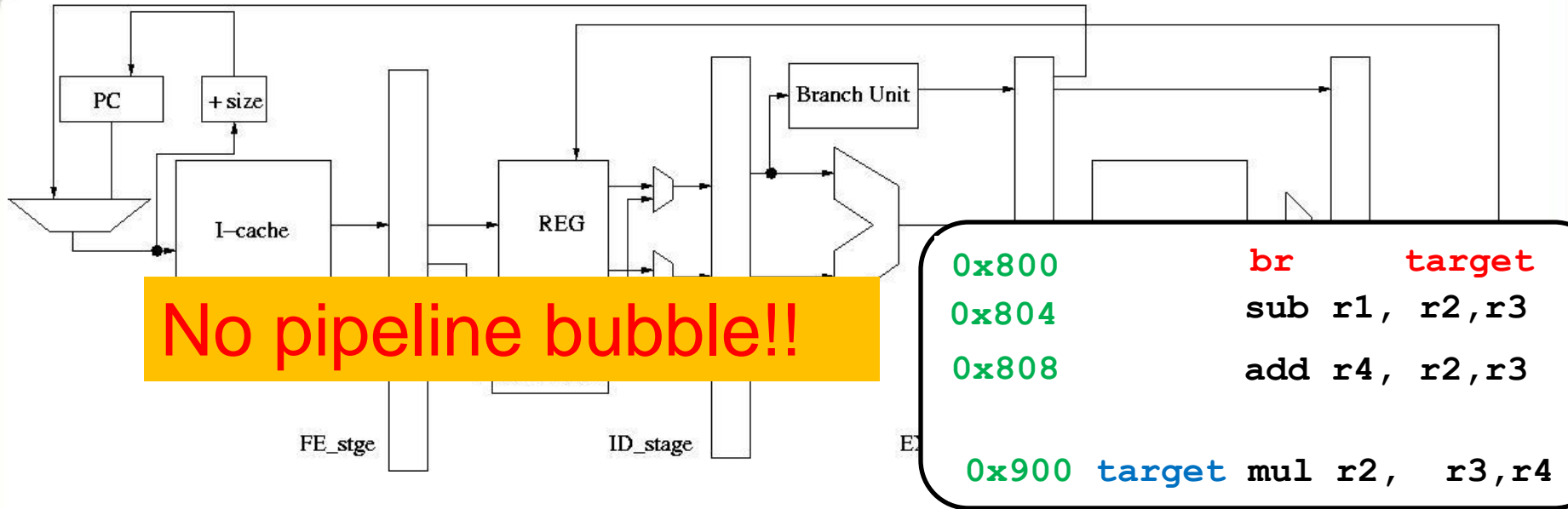
cycle	Fetch addr	FE	ID	EX	MEM	WB
1	0x800	br				
2	0x804	sub	br			
3	0x808	add	sub	br		
4	0x900	mul	add	sub	br	
5	0x904	div	mul	add	sub	br
6	0x908	add	div	mul	add	sub
7	0x90b	sub	add	div	mul	add

Delayed branch

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

- Different options:

Still Cancel or nullifying instructions

  – 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
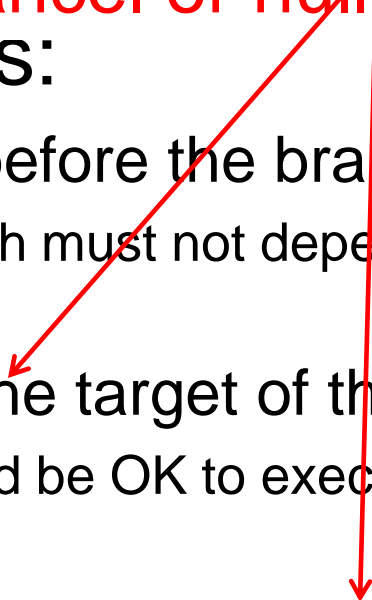
Branch and Branch Likely

- 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?

Delayed Slot in MIPS

```
0x0004000 lw     v0, $0004(v1) 1
0x0004004 jal    $08803000
0x0004008 addiu v0, v0, $0001
0x000400C sw     v0, $0004(v1) 3
0x0004010 jr ra
```

Function

```
0x00003000 lw    v2, $0008(v1)
0x00003004 jr    ra
0x00003008 add   v0, v0, v2
```

Init: v0 : 1 mem[v1+4] = 10
What will be v0 value at the beginning of the function?

What will be the return address?

Georgia Tech | College of Computing

Exception Handling in MIPS

- It is not allowed to report an exception for an instruction which could not be executed due to program control flow.

- For example, if a branch/jump is taken and the instruction after the branch is not to be executed, the address checks

- (alignment, MMU match/validity, access privilege) for that not-to-be-executed instruction may not generate any exception.

- ***Instruction fetch exceptions on branch delay-slots***

- For instructions occupying a branch delay-slot, any exceptions, including those generated by the fetch of that instruction,

- should report the exception results so that the branch can be correctly replayed upon return from the exception

- handler.

MIPS R10000

MIPS R10000

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

Shadow Registers

- Introduced from M4K RISC core
- Micro controllers:
  - Low core frequency but frequent executions
- To avoid GPR save/restore for fast interrupt support
- MIPS32 Release 2 architecture, including the M4K core, allow for up to eight copies of the GPRs to be present within the core itself.

Figure 1 MIPS32® Vectored Interrupt / Shadow Register Structure
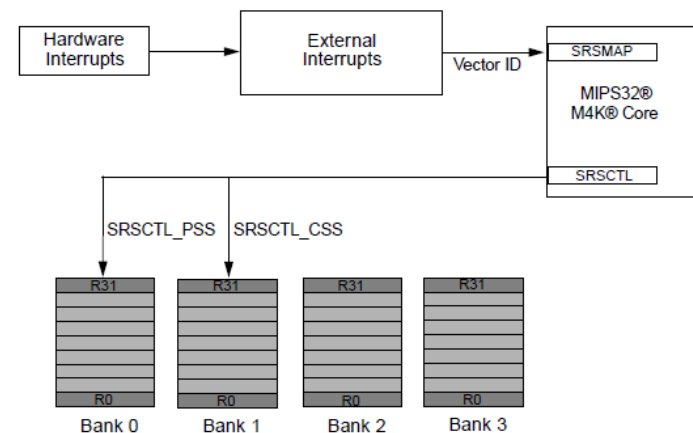
MIPS vs. ARM

	MIPS	ARM
# of registers	32	16
Branch	Delayed branch	predication
Context Switching cost	Full 32 Shadow registers (R10K)	2+6 registers
Function call	3 arguments + 2 return registers	Block copy operations
Pipeline complexity		
Starting point	Desktop processors → SOC	Soc → High-end

Update in Schedule

- 4/20 (W): Intel Atom processors
- 4/22 (F): Project progress meeting (each group 10 min)
- 4/25 (M): Phone/Tablet (mobile GPUs/GPGPUs)
- 4/27 (W): Project presentation
- 4/29 (F): Review for final exam