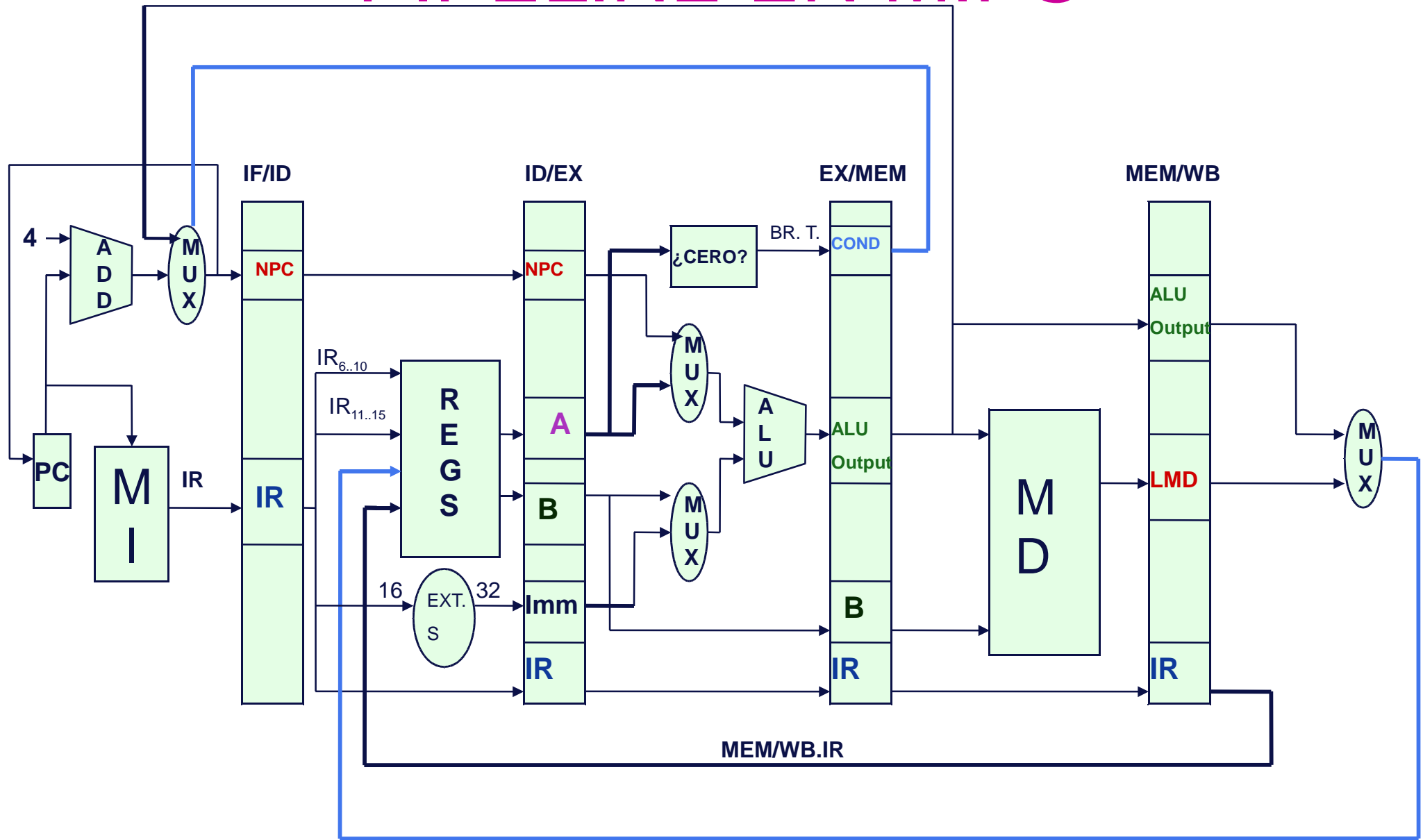


## APÉNDICE C\*

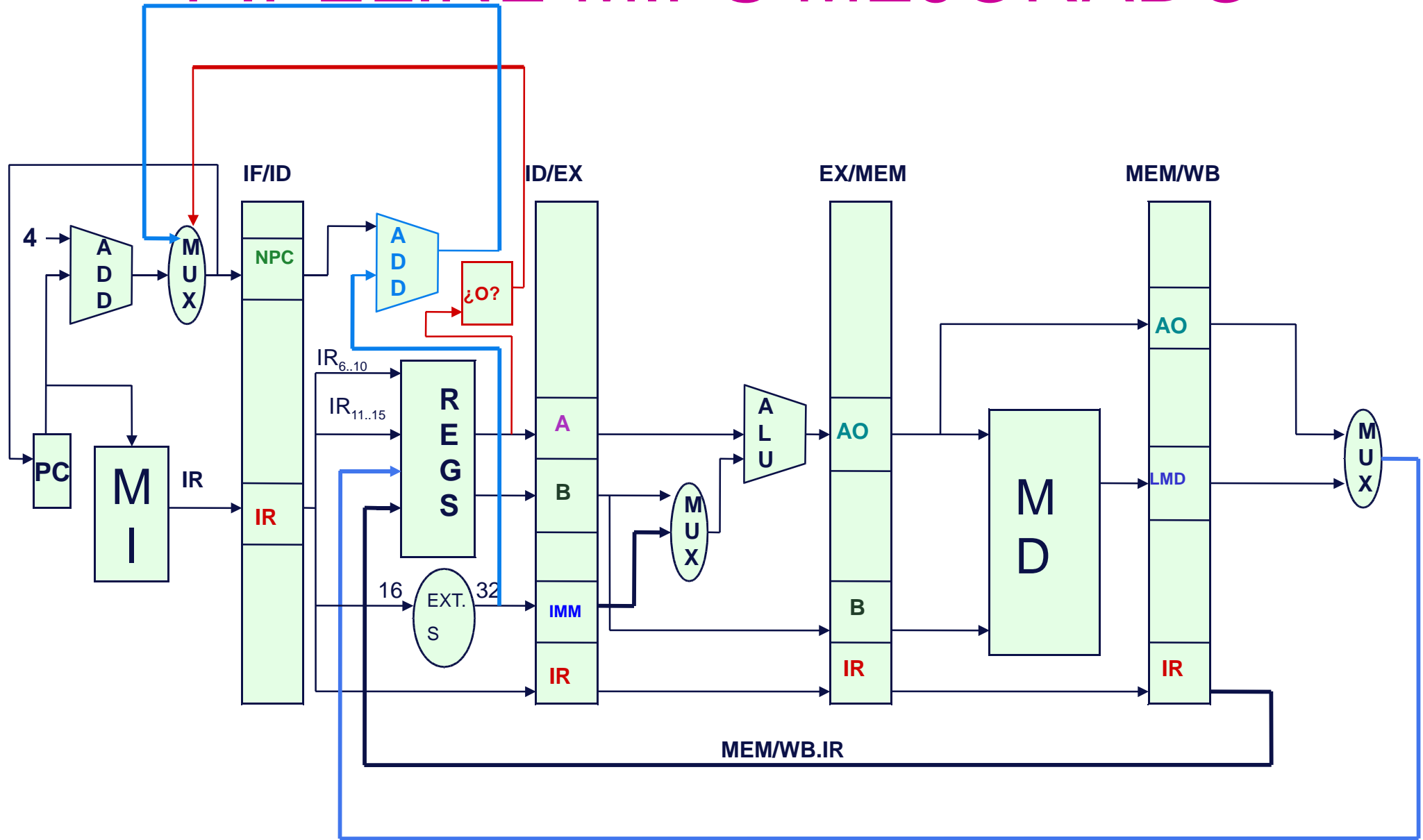
# “PIPELINING”: conceptos básicos e intermedios

\* Computer Architecture , Hennessy & Patterson, 5ta Edición

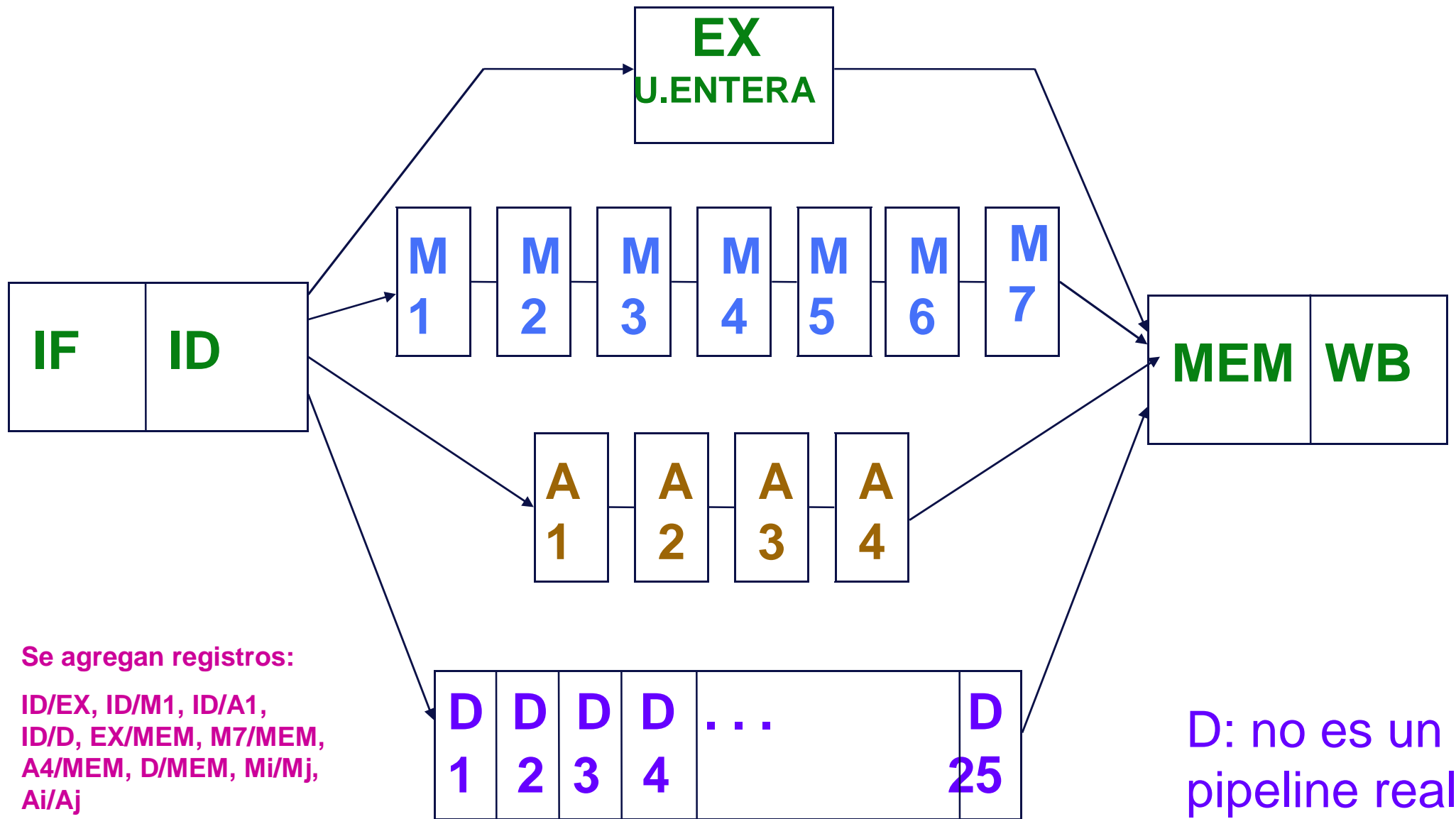
# PIPELINE EN MIPS



# PIPELINE MIPS MEJORADO



# PIPELINE MIPS (Fig. A.31)



# LATENCIA vs. INTERVALO DE REPETICIÓN

Latencia de unidad funcional:. Número de **ciclos** entre instrucción que **usa resultados** y la instrucción que **los produce**.

Intervalo de iniciación/ intervalo de repetición: número de **ciclos** que deben darse **entre la emisión** de instrucciones del mismo tipo.

UNIDAD FUNCIONAL	LATENCIA	INTERVALO INIC.
ALU Entera	0	1
Memoria datos (loads ent. y fp, 1 menos para latencia de store)	1	1
Sumador/Rest. FP	3	1
Multiplicador Ent./FP	6	1
Divisor Ent/FP	24	25

# EL PIPELINE DE LA MIPS R4000 (A.6)

Es realmente una máquina de 64 bits

- Pipelining muy profundo ([superpipelining](#)). 8 etapas:

- **IF** 1er parte fetch instrucc. selección PC, inicia inst. acceso a cache

- **IS** 2da parte fetch instrucc. Se completa acceso a cache

- **RF** Decod. Instruc, fetch registros, revisión conflictos, detección de si hubo hit en cache

- **EX** EFA, ALU op, branch target (dirección nueva) y cód. condición

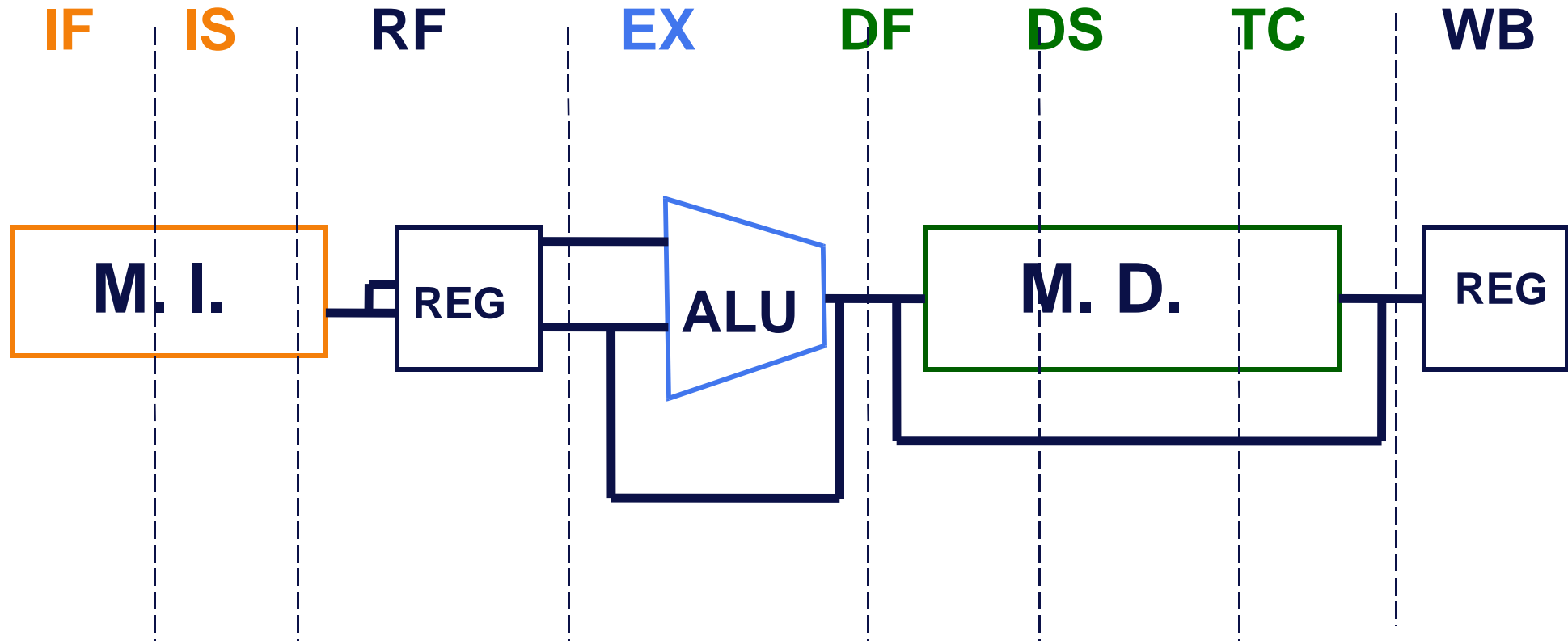
- **DF** Fetch de datos, 1era mitad acceso a cache de datos

- **DS** 2da mitad fetch de datos. Se completa acceso de datos en cache

- **TC** Revisión de etiqueta (tag), se determina si hubo hit en acceso a cache de datos

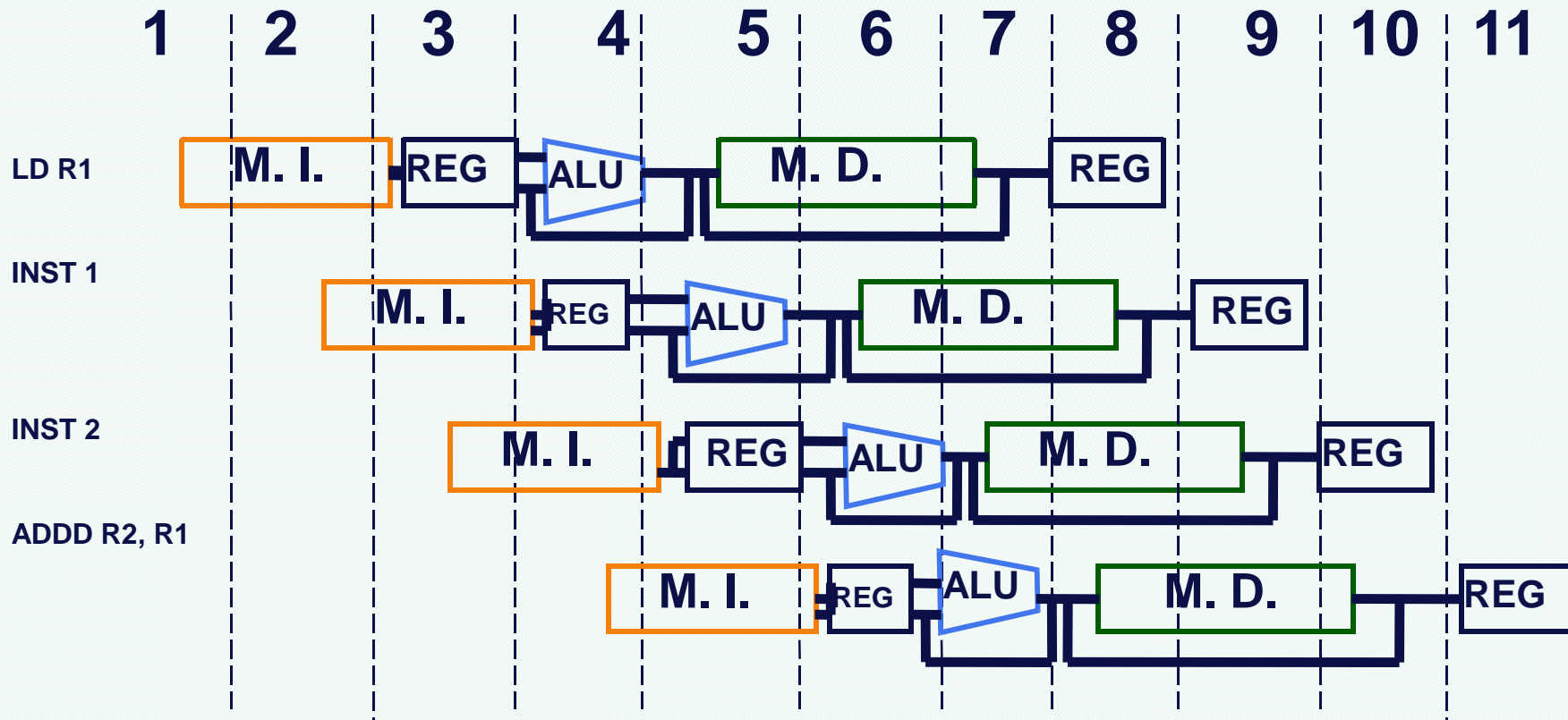
- **WB** “Write Back” para operaciones reg-reg y loads

# PIPELINE MIPS R4000 (CONT)



## Load tiene 2 ciclos de retraso-(delay slots)

=> debe haber 2 instrucciones o burbujas entre un load y una instrucción dependiente de ésta.

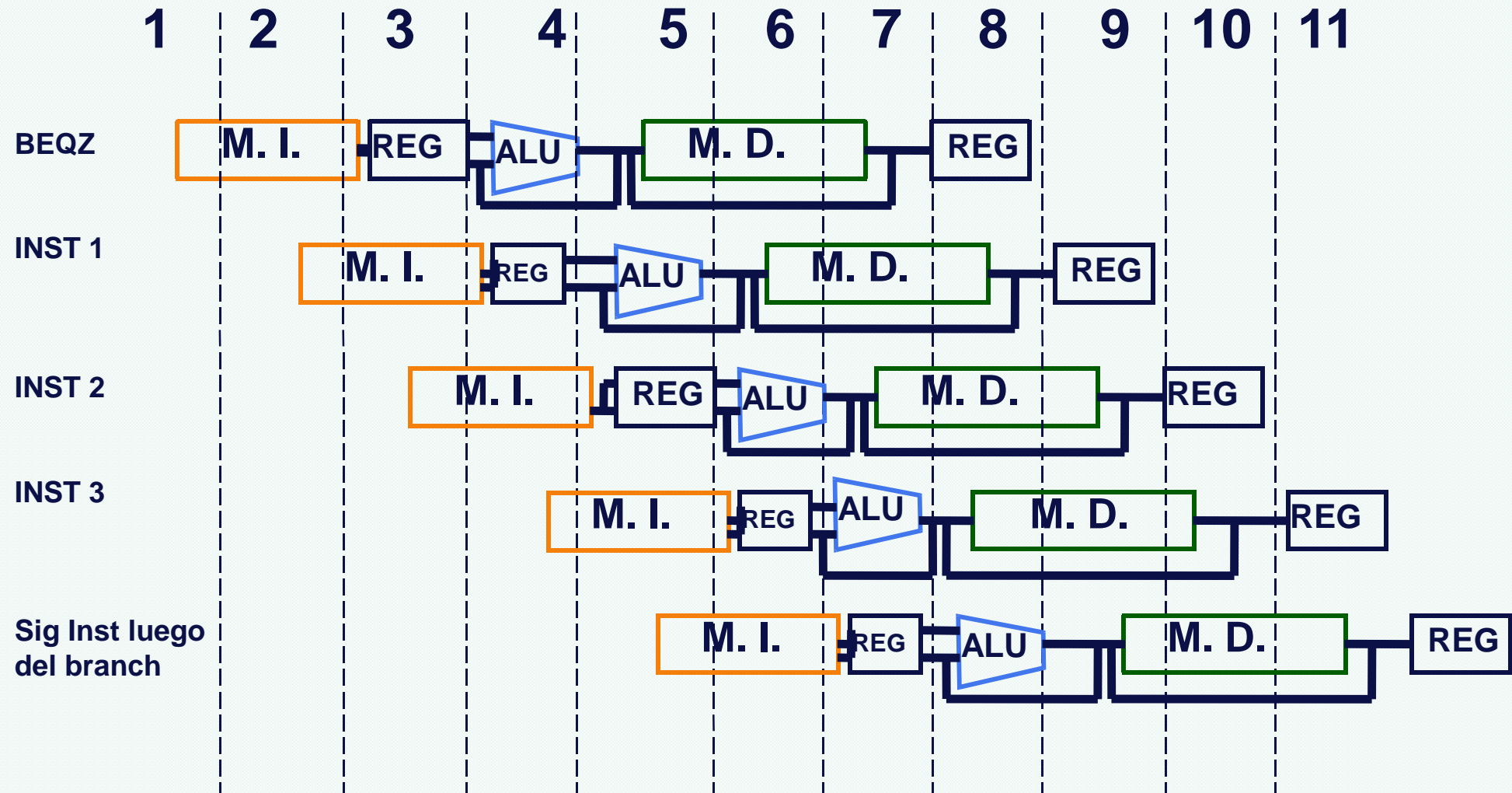


Nótese que el valor está disponible al final de DS del load y se hace un **forwarding** al ALU del ADD=> se usa **ANTES** de la revisión de etiqueta del caché (tag check). Por lo que si hay una falta en cache hay que “devolverse” y atrasarse un ciclo. (ocurre menos del 10% de las veces y es fácil detener instrucciones dependientes)



**BRANCH TIENE 3 CICLOS DE RETRASO (se resuelve en EX) =>  
CPI = 4 si no hay estrategia para disminuir retraso**

(Con branch retrasado se pueden incluir 3 instrucciones luego del branch:)



# MIPS SIEMPRE INCLUYE UNA INSTRUCCIÓN POR BRANCH RETRASADO y procesador hace predicción no tomado para las siguientes instrucciones:

(# 1 cuando Branch fue tomado, # 2 cuando Branch no fue tomado)

# 1	1	2	3	4	5	6	7	8	9
BRANCH	IF	IS	RF	EX	DF	DS	TC	WB	
DELAY SLOT		IF	IS	RF	EX	DF	DS	TC	WB
BRANCH + 2			IF	ID					
BRANCH + 3				IF					
B.TARGET					IF	IS	RF	EX	DF

# 2	1	2	3	4	5	6	7	8	9 ...
BRANCH	IF	IS	RF	EX	DF	DS	TC	WB	
DELAY SLOT		IF	IS	RF	EX	DF	DS	TC	WB
BRANCH + 2			IF	IS	RF	EX	DF	DS	TC ...
BRANCH + 3				IF	IS	RF	EX	DF	DS ...
BRANCH + 4					IF	IS	RF	EX	DF ...

# PIPELINE R4000 PARA PUNTO FLOTANTE

## 8 TIPOS DE ETAPAS

3 unidades funcionales: FP-DIV, FP-MUL, FP-ADD

Un sumador usado en paso final de MUL y DIV

ETAPA	UNIDAD FUNCIONAL	DESCRIPCION
A	FP-ADD	Suma mantisa
D	FP-DIV	Etapas de división
E	FP-MUL	Prueba de Excepción
M	FP-MUL	Primera etapa Multipl.
N	FP-MUL	Segunda etapa Multipl.
R	FP-ADD	Redondeo
S	FP-ADD	“Shift” de Operando
U	Las 3 unid. Func.	“Unpack” número FP

# OPERACIONES FP EN LA R4000

INST. FP	LATENCIA	INTERVALO INICIACION	SECUENCIA ETAPAS EN EL PIPE
Suma, resta	4	3	U, S + A, A + R, R + S
Multiplicación	8	4	U, E + M, M, M, M, N, N + A, R
División	36	35	U, A, R, D <sup>27</sup> , D + A, D + R D + R, D + A, D + R, A, R
Raíz Cuadrada	112	111	U, E, (A + R) <sup>108</sup> , A, R
Negación	2	1	U, S
Valor Absoluto	2	1	U, S
Comparación FP	3	2	U, A, R