CLASE 6
MEJORAS EN LA ORGANIZACIÓN
Arquitectura Harvard
Segmentación del ciclo de instrucción

Mejoras del REPERTORIO DE INSTRUCCIONES (ISA)

- Aumento de la cantidad de memoria
- Aumento del número de registros de propósitos generales: disminución de los accesos a memoria.
- Registros de uso específico
- Mejora de la ALU: operaciones y tipos de datos
- Ampliación del repertorio de saltos condicionales
- Loops: mejoras en las repeticiones
- Nuevos modos de direccionamiento y registros de uso específico: mejora en el acceso a los datos
- La pila: mejora en la implementación de subrutinas
- El sistema de entrada/salida, interrupciones

Código compacto: Ahorro de memoria de programa. Captación y decodificación de menos instrucciones. El flujo de datos es el mismo.
Código ordenado: Importante si se programa manualmente en assembler. No tan importante si la tarea la realiza un compilador.
Mejora de la performance? No necesariamente. No es fácil de evaluar pues tiene múltiples variables interrelacionadas.

CLASE DE HOY: mejoras de la ORGANIZACIÓN del procesador

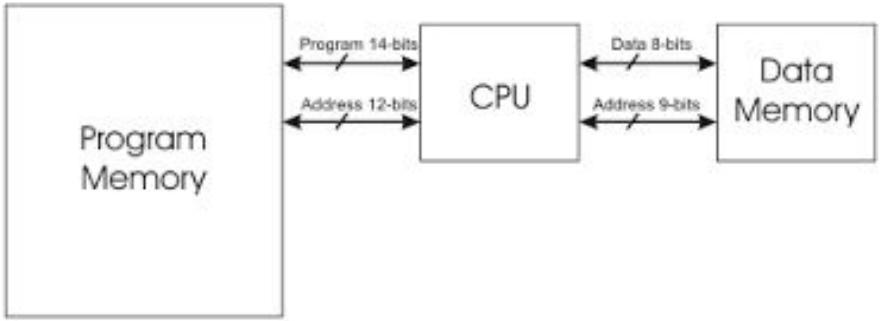
Optimización del ciclo de instrucción. Paralelismo. Mayor productividad.

- Arquitectura Harvard
- Segmentación

Las mejoras en Organización continúan en próximas materias: Jerarquía del subsistema de memoria, procesadores superescalares, procesadores multinúcleo, etc.

PROXIMA CLASE: mejoras en la TECNOLOGÍA DE IMPLEMENTACIÓN del procesador

Arquitectura Harvard

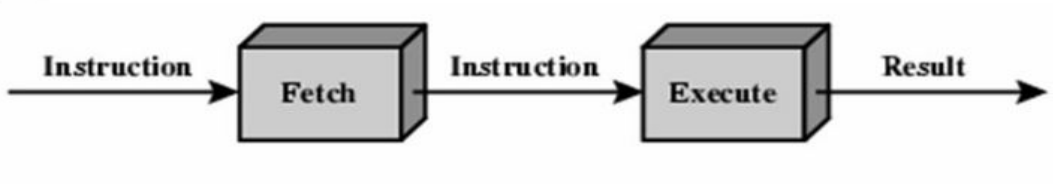


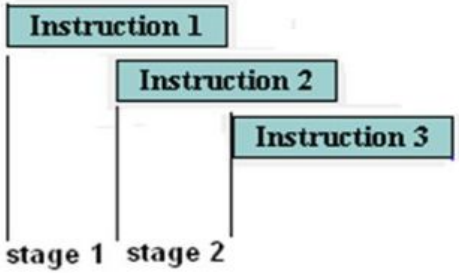
Microchip PIC (1993) 8-bit RISC
Acumulador (W)
File RAM: 8 x 368
Memoria de programa: 14 x 4K
4 ciclos por instrucción:
- Q1: Fetch the current instruction
- Q2: Read the data for current instruction
- Q3: Execute the current instruction (ie, add/shift/subtract/test data)
- Q4: Write any data for the current instruction

La arquitectura Harvard utiliza memorias físicamente separadas para instrucciones y datos. Requiere buses dedicados para cada una de ellas. Recordar que la arquitectura von Neumann utiliza un único bus y un único espacio de memoria, donde conviven los datos y las instrucciones, por lo tanto en cada ciclo de instrucción hay que acceder dos veces a la memoria a través de un único bus.

Ventajas:
Acceso simultáneo a instrucciones y sus operandos.
Ambas memorias pueden tener diferentes dimensiones.
Menos chances de corrupción del programa.
Soluciona el cuello de botella (bottleneck) que presenta la memoria en la arquitectura de von Neumann.
Desventajas:
La memoria de programa sobrante no se puede utilizar para datos (y viceversa).
El programa no puede auto-escribirse.
CPU más costosa. Dos buses.

Segmentación de instrucciones
(Instruction pipelining)





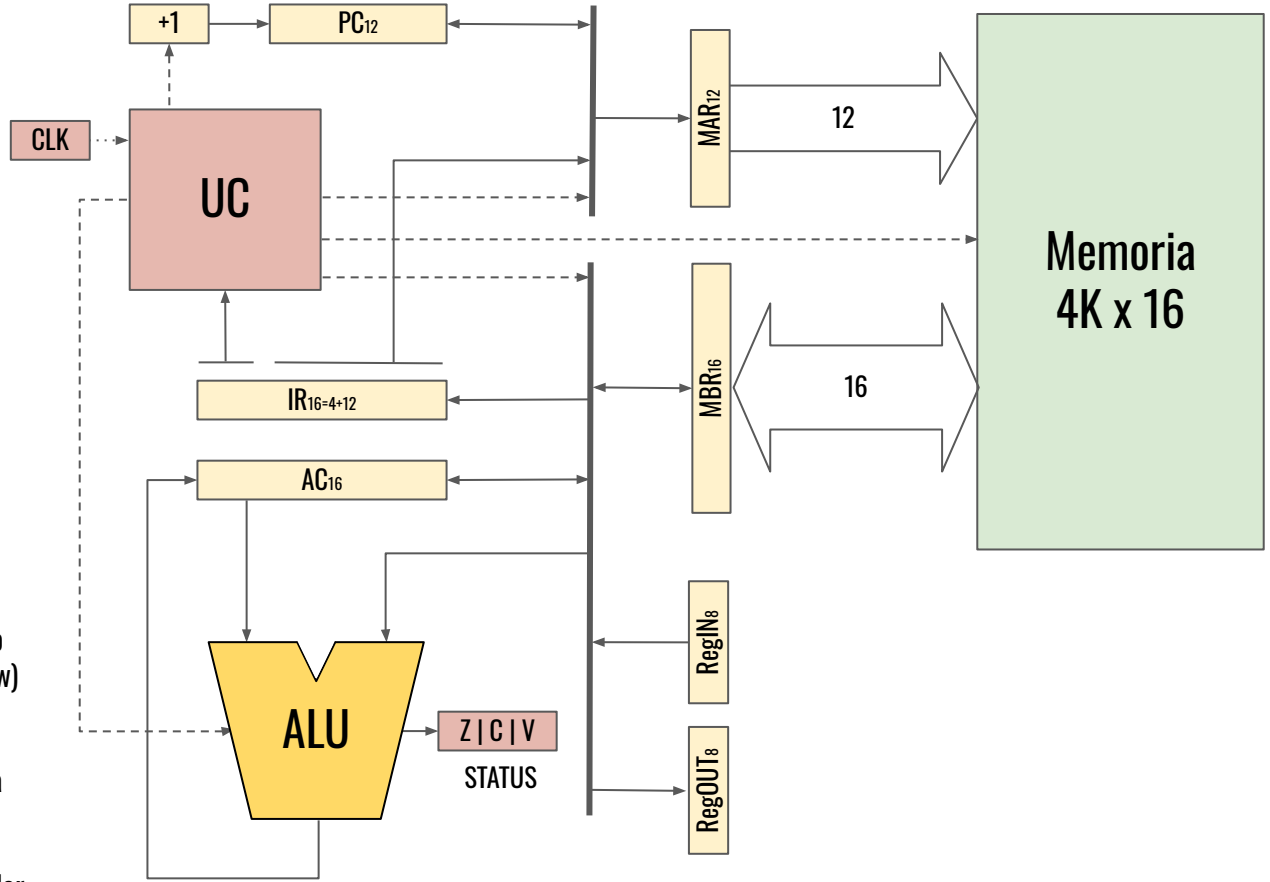
La segmentación es una técnica de diseño que intenta tener ocupadas con instrucciones todas las partes del procesador dividiendo las instrucciones en una serie de pasos secuenciales que efectuarán distintas unidades de la CPU, tratando en paralelo las diferentes partes de la instrucción.
La técnica es más eficiente en algunos diseños que en otros. Harvard ayuda manteniendo separadas las instrucciones de los datos.
-> ISA RISC, mucho más fácil de segmentar (Instrucciones de largo fijo, load/store y muchos registros)

```
0011) ADD X

      MAR <- PC
      MBR <- M[MAR]
      IR  <- MBR
      PC  <- PC + 1

      MAR <- IR(11-0)
      MBR <- M[MAR]

      AC  <- AC + MBR
```

MARIE

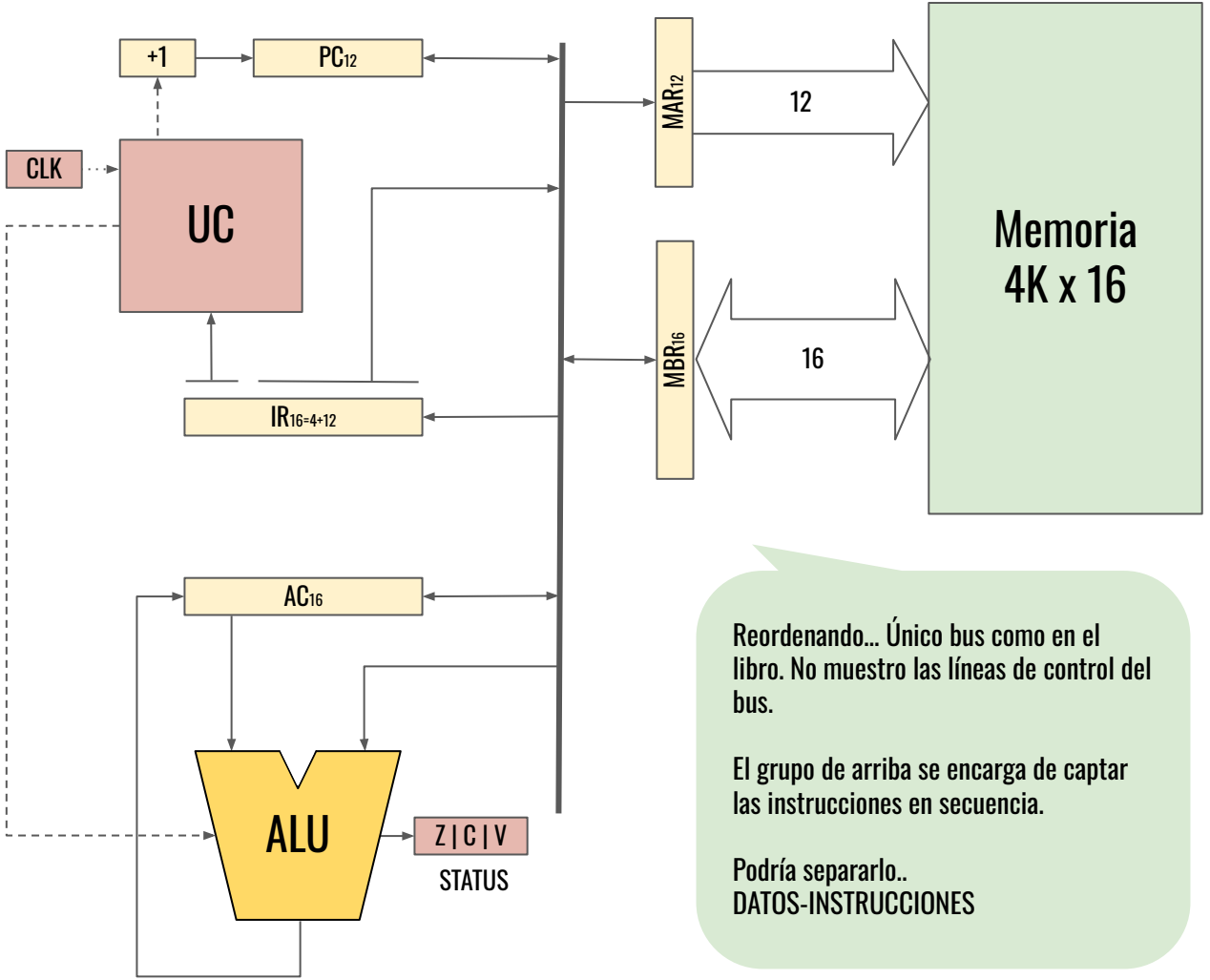
La suma requiere 7 “pasos” (ciclos de reloj). Se puede reducir a 6 (PC+1 en paralelo). Pero no menos porque hay una secuencia (dataflow) que respetar.

Se accede dos veces a memoria, una vez para captar la instrucción y otra para captar el operando indicado en la instrucción. Finalmente realiza la suma entre el acumulador y el operando.



0011) ADD X

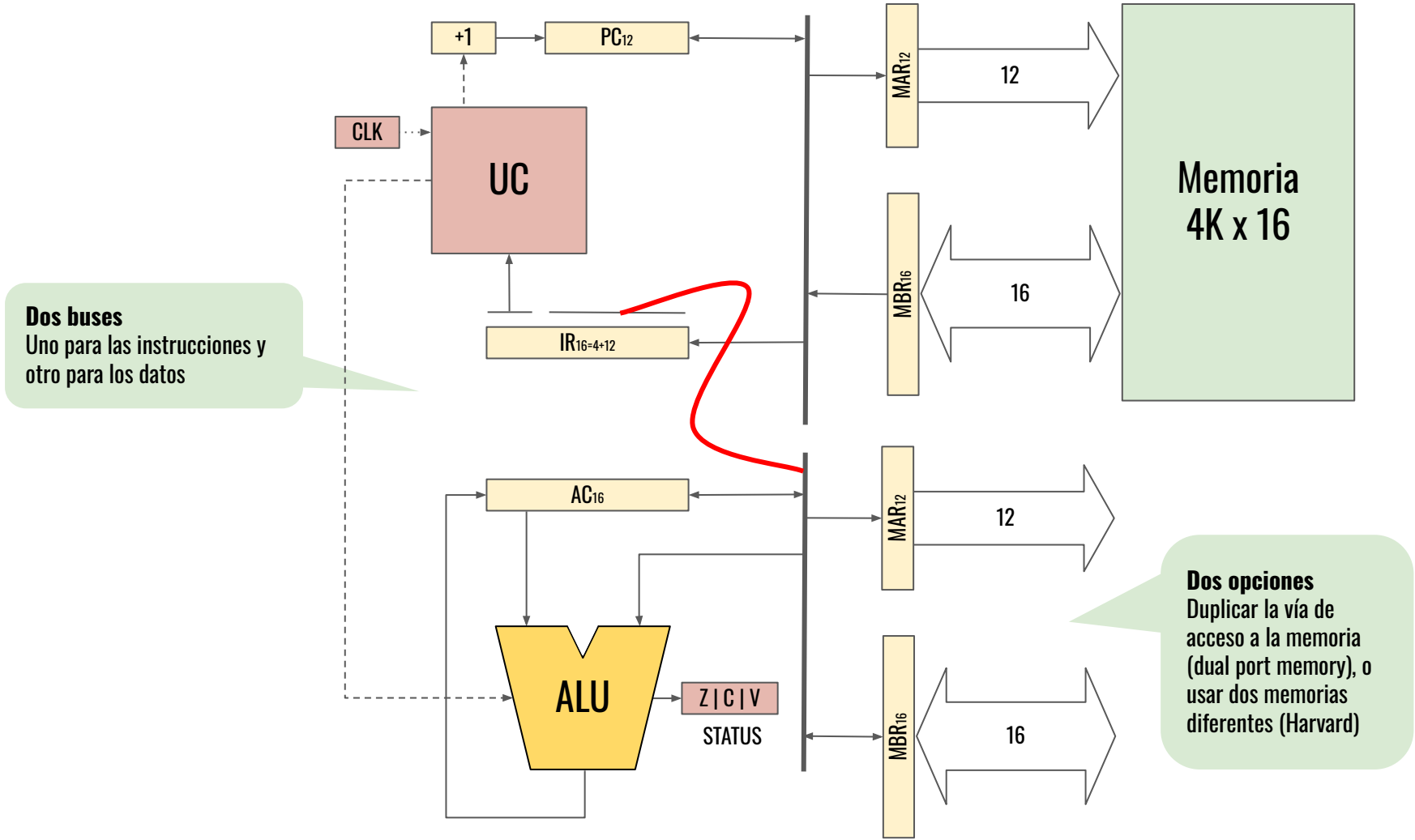
```
MAR <- PC
MBR <- M[MAR]
IR  <- MBR
PC  <- PC + 1

MAR <- IR(11-0)
MBR <- M[MAR]

AC  <- AC + MBR
```



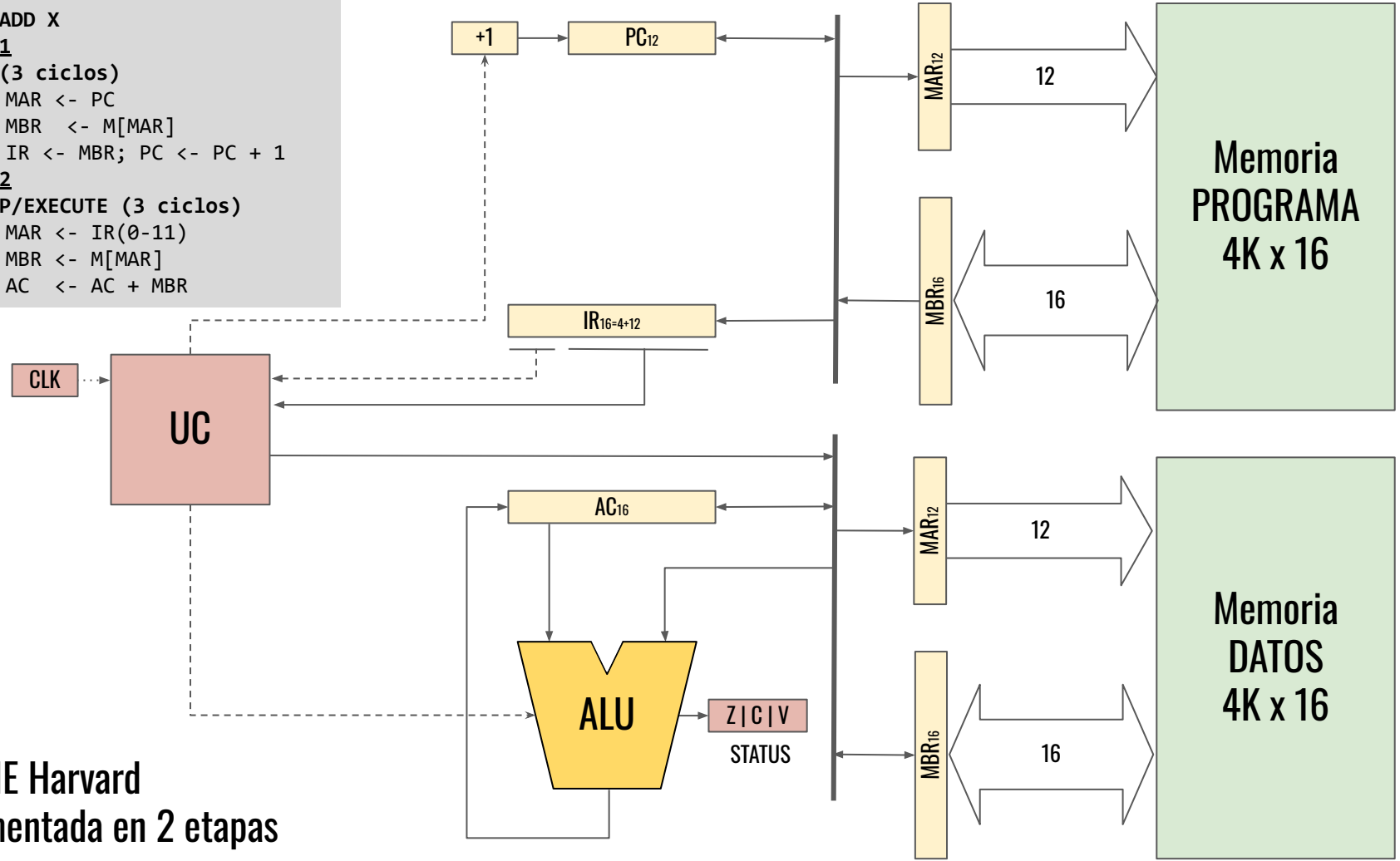
Reordenando... Único bus como en el libro. No muestro las líneas de control del bus.

El grupo de arriba se encarga de captar las instrucciones en secuencia.

Podría separarlo..
DATOS-INSTRUCCIONES

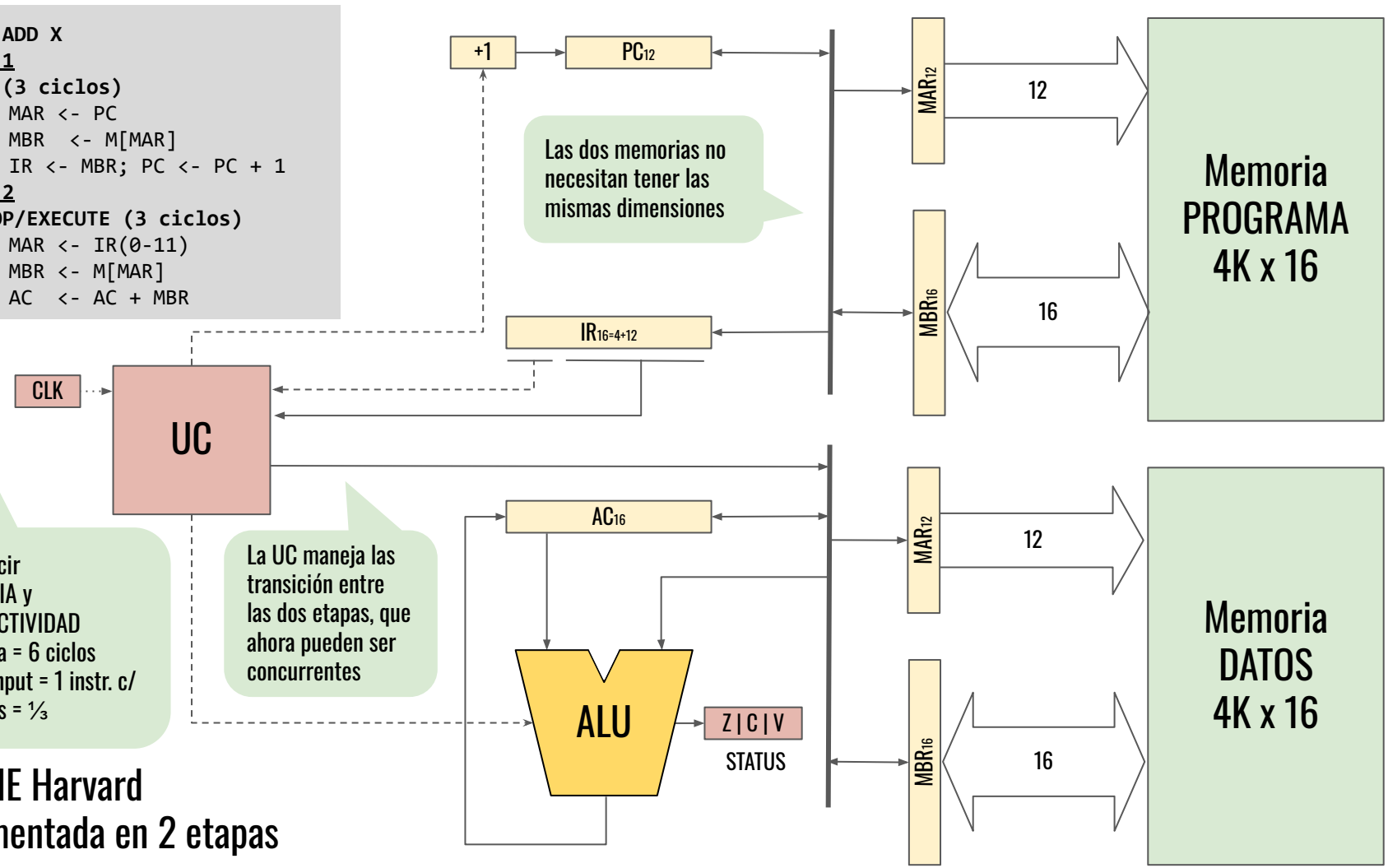

+1
PC12
CLK
UC
IR16=4+12
MAR12
12
MBR16
16
Memoria
4K x 16
AC16
ALU
Z | C | V
STATUS
MAR12
12
MBR16
16
Dos buses
Uno para las instrucciones y otro para los datos
Dos opciones
Duplicar la vía de acceso a la memoria (dual port memory), o usar dos memorias diferentes (Harvard)

```
0011) ADD X
ETAPA 1
FETCH (3 ciclos)
       MAR <- PC
       MBR  <- M[MAR]
       IR <- MBR; PC <- PC + 1
ETAPA 2
FETCHOP/EXECUTE (3 ciclos)
       MAR <- IR(0-11)
       MBR <- M[MAR]
       AC  <- AC + MBR
```



MARIE Harvard
Segmentada en 2 etapas

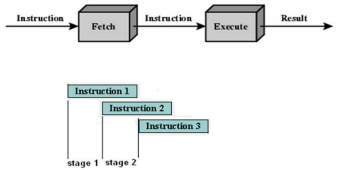
```
0011) ADD X
ETAPA 1
FETCH (3 ciclos)
       MAR <- PC
       MBR  <- M[MAR]
       IR <- MBR; PC <- PC + 1
ETAPA 2
FETCHOP/EXECUTE (3 ciclos)
       MAR <- IR(0-11)
       MBR <- M[MAR]
       AC  <- AC + MBR
```

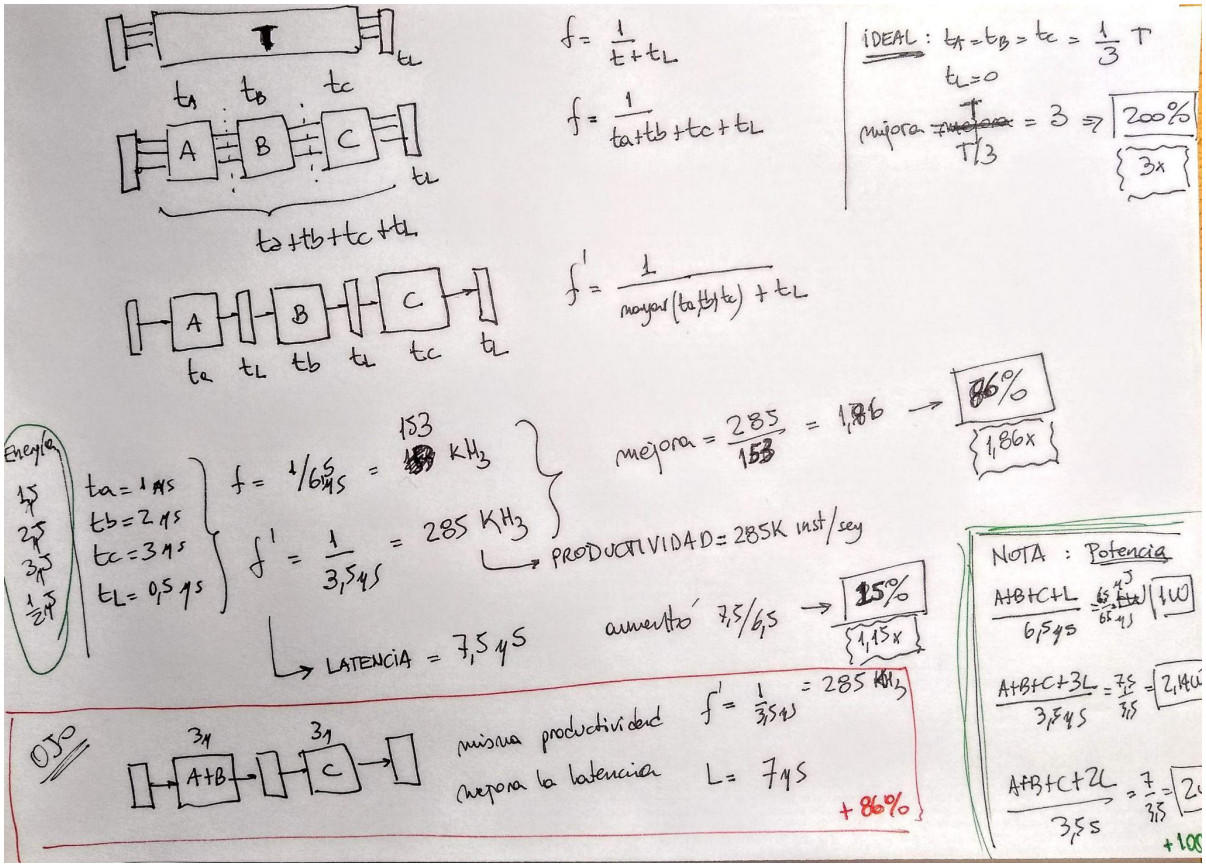


MARIE Harvard
Segmentada en 2 etapas

Latencia y productividad - PIZARRÓN

La segmentación mejora la productividad, desmejora la latencia y aumenta la potencia. ¿Por qué?





En este ejemplo, una mejora de productividad del 86% implica un aumento de la latencia del 15% y un aumento de la potencia del 100%.

0011) ADD X

FETCH/DECODE 2->1

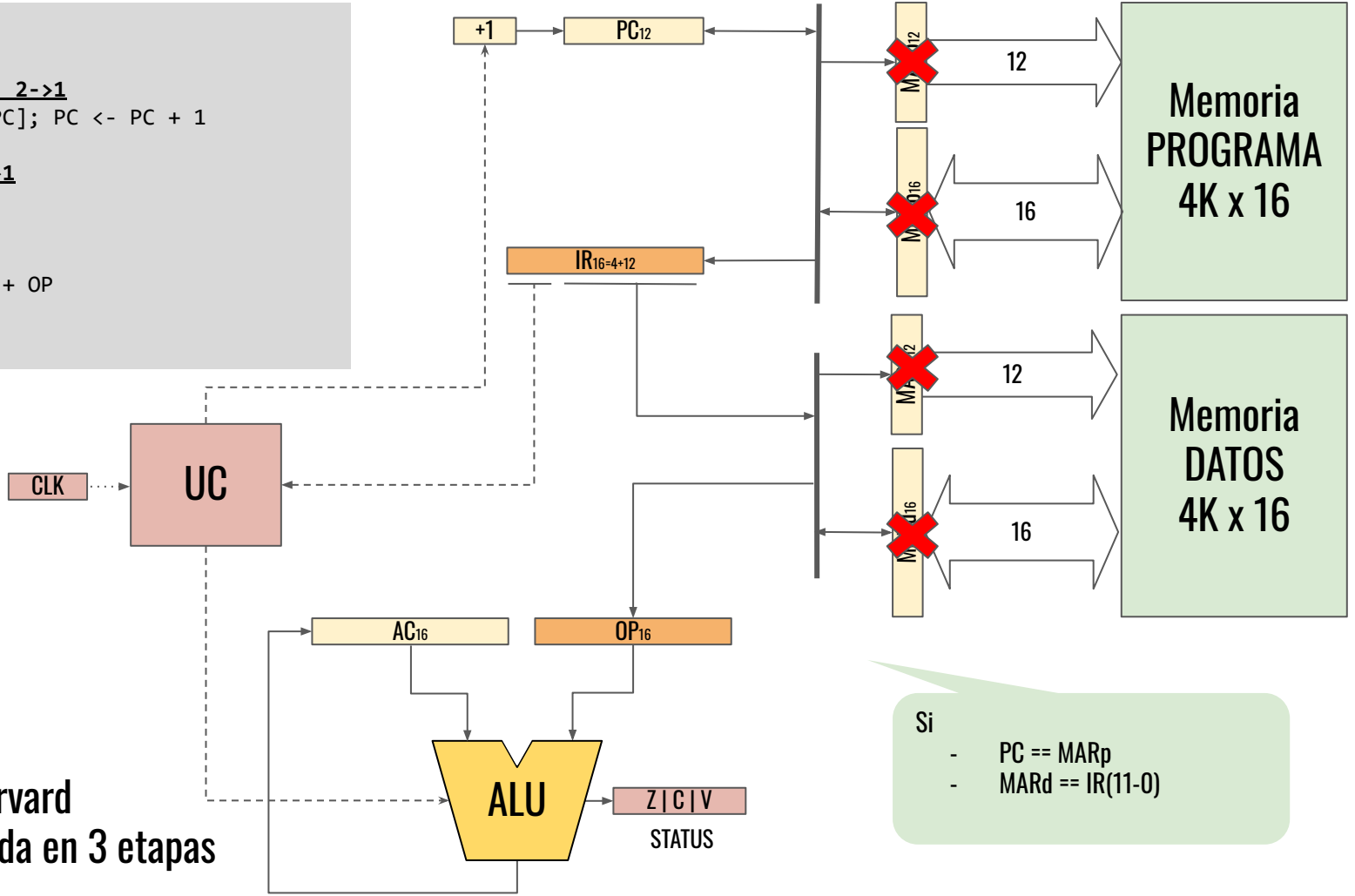
```
IR  <- M[PC]; PC <- PC + 1
```

FETCHOP 2->1

```
OP <- M[X]
```

EXECUTE 1

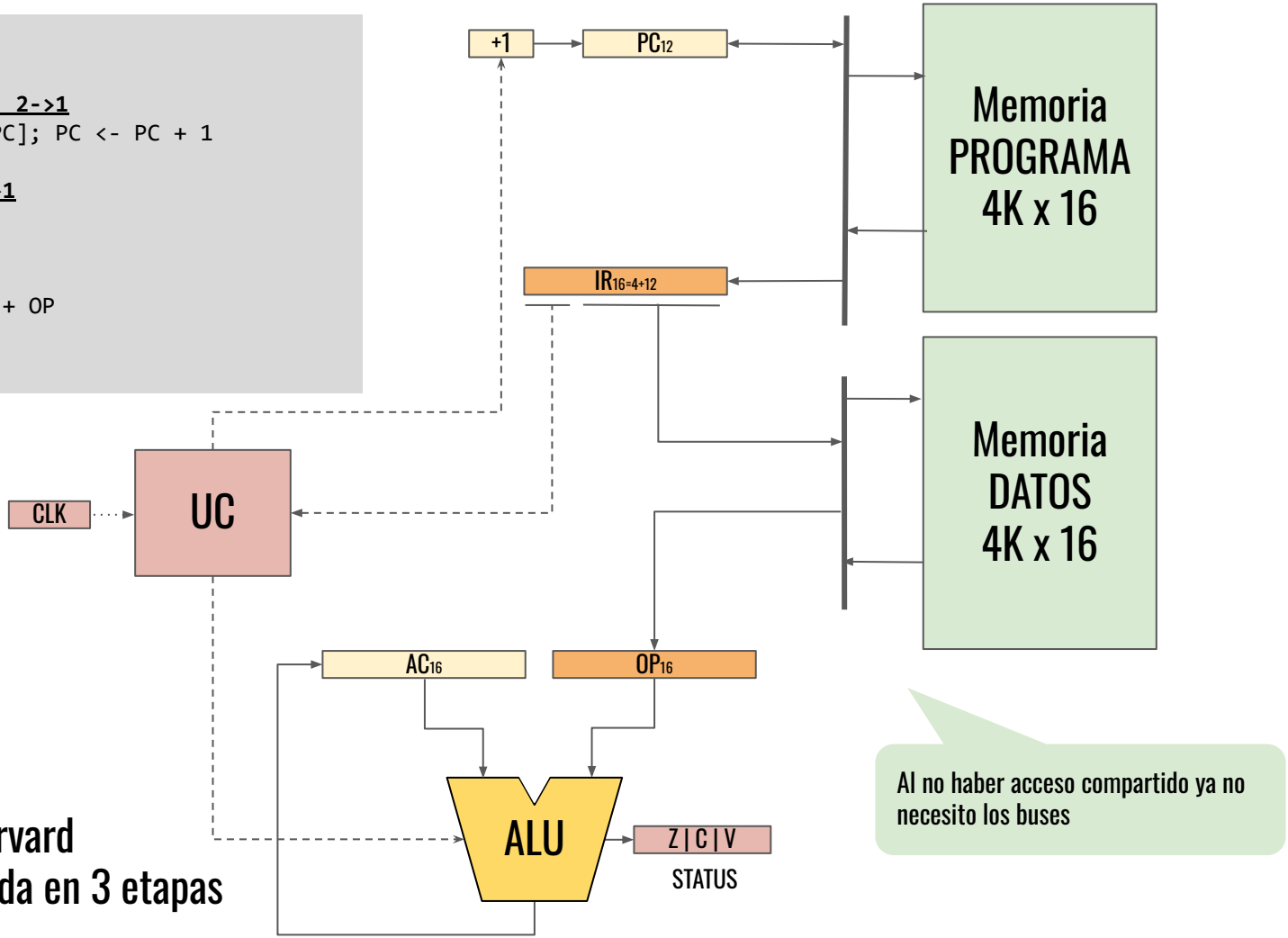
```
AC  <- AC + OP
```



MARIE Harvard
Segmentada en 3 etapas

Si
- PC == MARp
- MARd == IR(11-0)

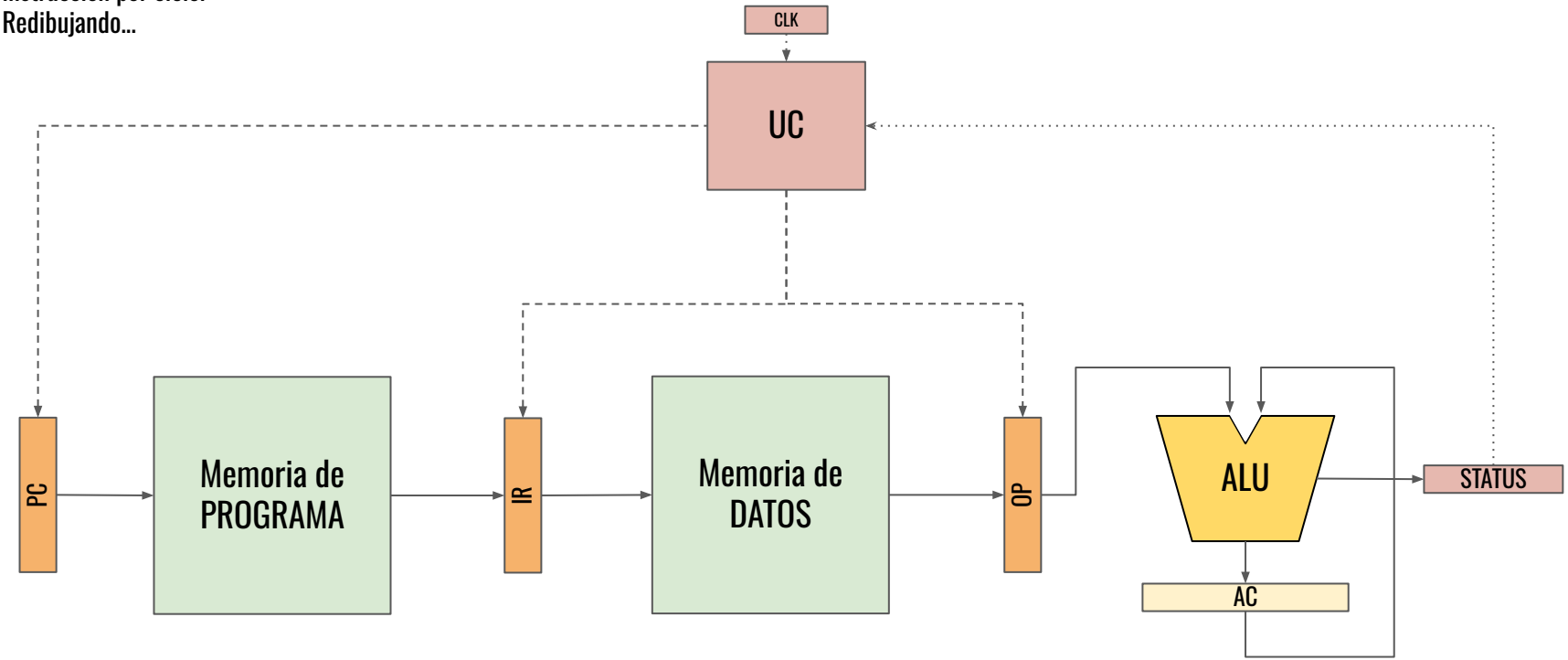
```
0011) ADD X

FETCH/DECODE  2->1
   IR  <- M[PC]; PC <- PC + 1

FETCHOP   2->1
   OP <- M[X]

EXECUTE   1
   AC  <- AC + OP
```



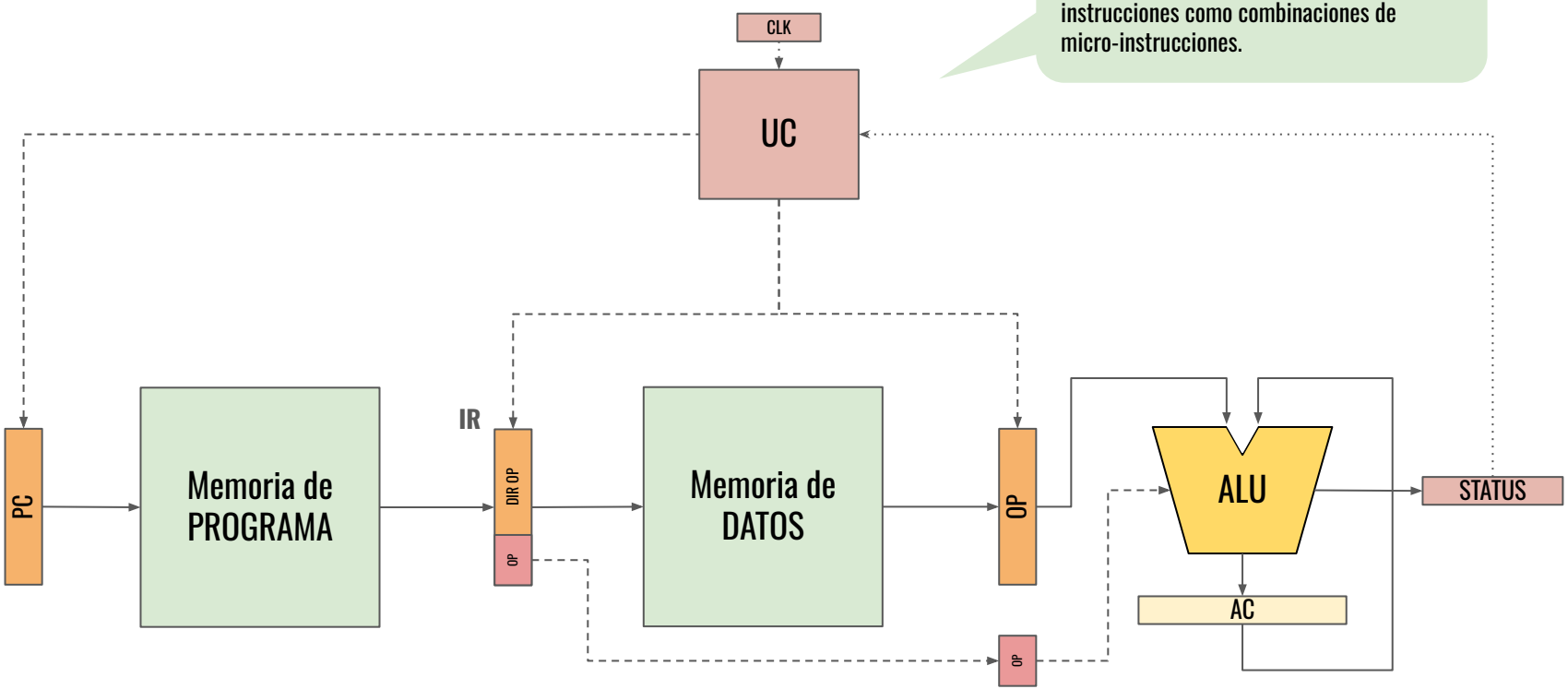
MARIE Harvard
Segmentada en 3 etapas

Al no haber acceso compartido ya no necesito los buses

Entonces la productividad puede llegar a una instrucción por ciclo.
Redibujando...



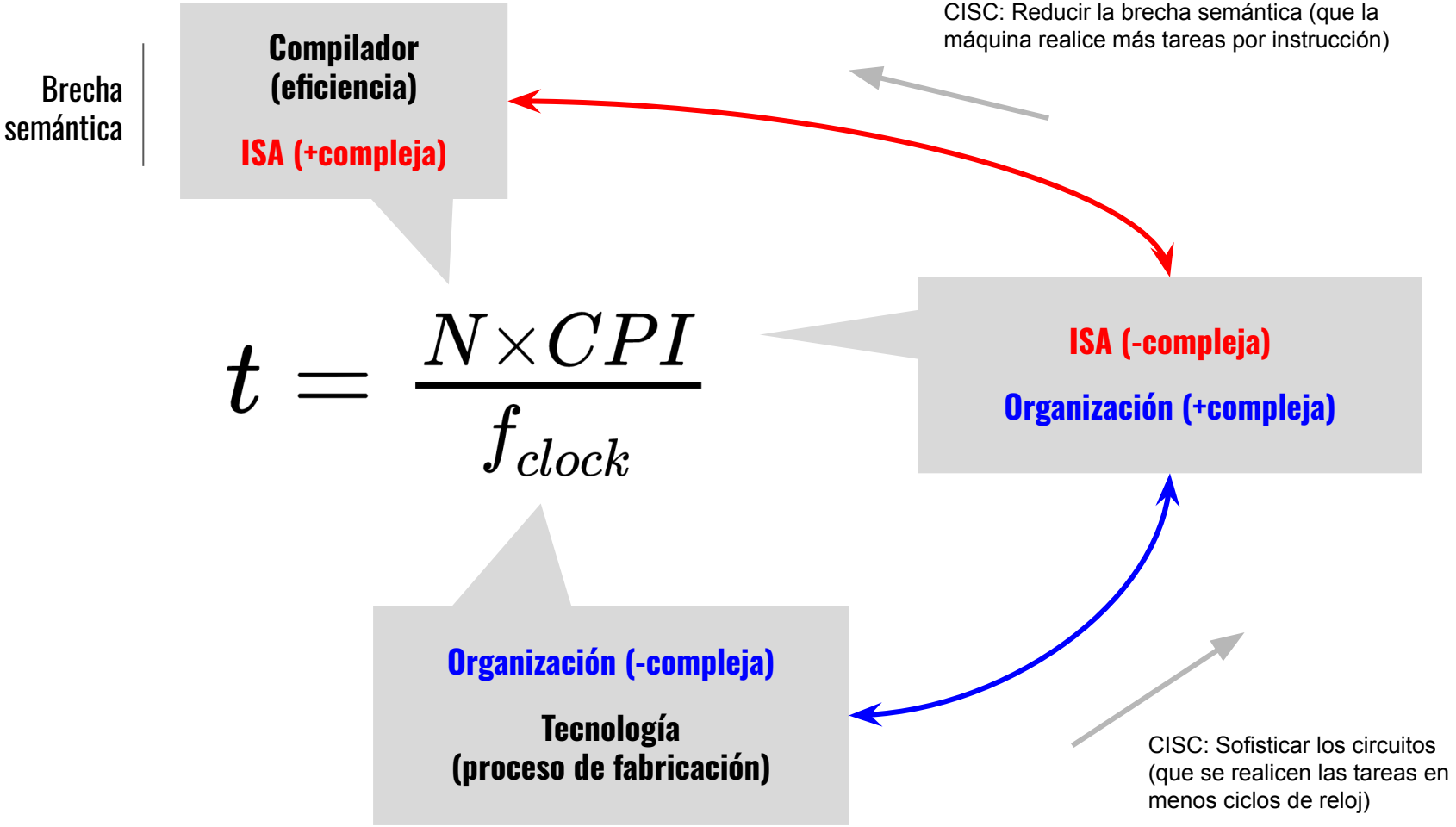
UC Cableada (hardwired)
Se pierde la capacidad de agregar nuevas instrucciones como combinaciones de micro-instrucciones.



Analizar LATENCIA y PRODUCTIVIDAD (throughput) en general
Latencia empeora (aumenta)
Throughput mejora (aumenta)

Continuará…
En Arquitectura Avanzada de Procesadores

Organización y Arquitectura


Brecha semántica
Compilador (eficiencia)
ISA (+compleja)
CISC: Reducir la brecha semántica (que la máquina realice más tareas por instrucción)
$t = \frac{N \times CPI}{f_{clock}}$
ISA (-compleja)
Organización (+compleja)
Organización (-compleja)
Tecnología (proceso de fabricación)
CISC: Sofisticar los circuitos (que se realicen las tareas en menos ciclos de reloj)

Un problema típico de organización

Se dispone de una versión del procesador MARIE que utiliza una segmentación en **dos etapas**, en la cual todas las instrucciones toman 3 ciclos de reloj (por etapa). Suponga que está implementado con una determinada tecnología CMOS, por ejemplo 120 nm, en la cual se verifica que el procesador puede operar con una frecuencia de reloj máxima de 100 MHz.

Se propone, a modo de mejora, utilizar una segmentación en **tres etapas**. Se trataría de una nueva implementación de MARIE, en la cual todas las instrucciones tomarían 1 ciclo de reloj, excepto los saltos que tomarían 2 ciclos. Suponga, en principio, que el programa contiene un 30% de instrucciones de salto. Se prevé que, utilizando la misma tecnología de implementación de 120 nm, esta nueva organización podría funcionar con un reloj de 60 MHz como máximo, debido a la organización más compleja.

La nueva propuesta, ¿es realmente una mejora? ¿Cómo la expresaría? Si en la versión original un programa tardaba 10 segundos en ejecutarse, ¿cuánto tardaría en la nueva versión?

Un problema típico de organización

Se dispone de una versión del procesador MARIE que utiliza una segmentación en **dos etapas**, en la cual todas las instrucciones toman 3 ciclos de reloj (por etapa). Suponga que está implementado con una determinada tecnología CMOS, por ejemplo 120 nm, en la cual se verifica que el procesador puede operar con una frecuencia de reloj máxima de 100 MHz.

Se propone, a modo de mejora, utilizar una segmentación en **tres etapas**. Se trataría de una nueva implementación de MARIE, en la cual todas las instrucciones tomarían 1 ciclo de reloj, excepto los saltos que tomarían 2 ciclos. Suponga, en principio, que el programa contiene un 30% de instrucciones de salto. Se prevé que, utilizando la misma tecnología de implementación de 120 nm, esta nueva organización podría funcionar con un reloj de 60 MHz como máximo, debido a la organización más compleja.

La nueva propuesta, ¿es realmente una mejora? ¿Cómo la expresaría? Si en la versión original un programa tardaba 10 segundos en ejecutarse, ¿cuánto tardaría en la nueva versión?

Solución:

$$N_1 = N_2 \quad \textit{(no hay cambios en la ISA)}$$

$$CPI_1 = 3 \qquad CPI_2 = 0.7*1 + 0.3*2 = 1.3$$

$$f_1 = 100\ MHz \qquad f_2 = 60\ MHz$$

$$t_1 * 100/3 = t_2 * 60/1.3$$

$$t_2 = 0.72\ t_1$$

$$mejora = t_1/t_2 = 1.38$$

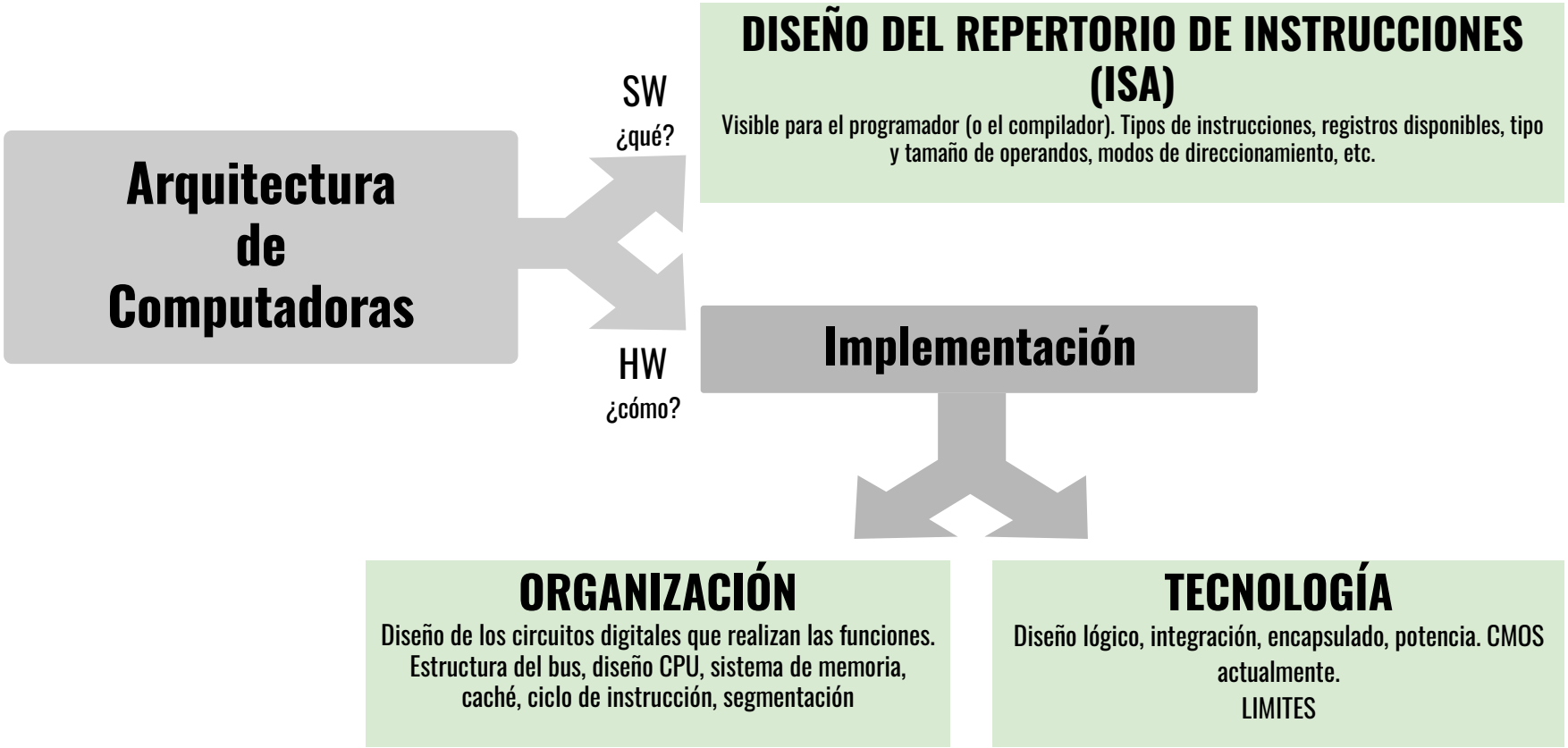
$$t = \frac{N \times CPI}{f_{clock}}$$

El repertorio de instrucciones y la tecnología no parecen jugar un rol en este problema. Es correcto decir eso?

El compilador podría ser importante: en la nueva arquitectura una solución con menos saltos sería más eficiente. En la anterior no era relevante (todas las instrucciones tardaban lo mismo).

El programa tardará *7.2 s* en ejecutarse en la máquina mejorada ¿está bien decir que el procesador mejoró un 38%?

¿Qué valor de f_2 hace que la propuesta ya no signifique una mejora?
Si el porcentaje de saltos pudiera reducirse a 20%, ¿cuál sería la frecuencia mínima de operación?


Arquitectura de Computadoras
SW
¿qué?
HW
¿cómo?
DISEÑO DEL REPERTORIO DE INSTRUCCIONES (ISA)
Visible para el programador (o el compilador). Tipos de instrucciones, registros disponibles, tipo y tamaño de operandos, modos de direccionamiento, etc.
Implementación
ORGANIZACIÓN
Diseño de los circuitos digitales que realizan las funciones. Estructura del bus, diseño CPU, sistema de memoria, caché, ciclo de instrucción, segmentación
TECNOLOGÍA
Diseño lógico, integración, encapsulado, potencia. CMOS actualmente.
LIMITES

ALGORITMO

Diseño del compilador

Arquitectura de Computadoras

SW

HW

DISEÑO DEL REPERTORIO DE INSTRUCCIONES (ISA)

Visible para el programador (o el compilador). Tipos de instrucciones, registros disponibles, tipo y tamaño de operandos, modos de direccionamiento, etc.

Implementación

Ejemplo: DSP
¿dispone de MAC o MUL? (ISA)
¿cuántos ciclos demoran? (ORG);
¿cuál es la máxima frecuencia de trabajo y el consumo de potencia? (TEC)

ORGANIZACIÓN

Diseño de los circuitos digitales que realizan las funciones. Estructura del bus, diseño CPU, sistema de memoria, caché, ciclo de instrucción, segmentación

TECNOLOGÍA

Diseño lógico, integración, encapsulado, potencia. CMOS actualmente.
LIMITES

Repertorio de instrucciones (ISA)

Organización

*"La arquitectura de computadoras, como otras arquitecturas, es el arte de determinar las **necesidades del usuario de una estructura** y luego **diseñar la estructura** para satisfacer dichas necesidades tan eficientemente como sea posible dentro de ciertas **limitaciones económicas y tecnológicas**".*

Frederick P. Brooks, IBM, 1962.

Tecnología

Ejemplo: Intel Tick–Tock model

Estrategia desde 2007: un cambio en la organización (tock) seguido por una mejora tecnológica (tick).
Hace 40 años que no cambian el ISA.

ERROR COMÚN

Suponer que dos procesadores con idéntica ISA se pueden comparar por su frecuencia de reloj

OTRAS DISCUSIONES

Computadora óptica (cambiando la tecnología de implementación se pueden mantener la ISA y la ORG)
AMD vs. Intel (misma ISA y TEC, cambia ORG)

Intel 8051

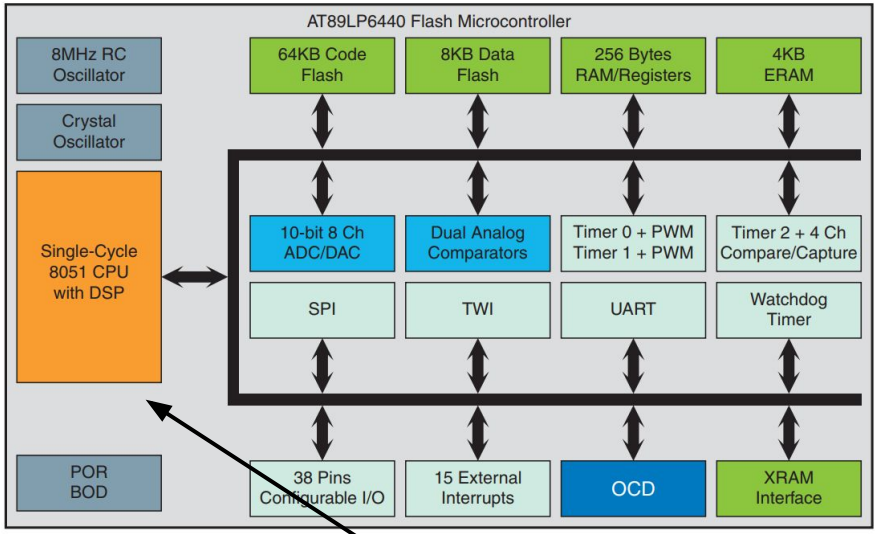
La arquitectura 8051 ha sido utilizada en la industria durante décadas, por lo que se mantiene popular hoy en día por su disponibilidad de bibliotecas y herramientas. El diseño original (1980) tomaba 12 ciclos de reloj (x1, x2, x3 o x4) por instrucción.

MSC®51 instruction set
8-bit data, 16-bit address, Harvard, **boolean**
Acc + 32 registros (4 bancos de 8)

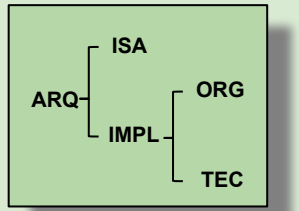
Hoy mismo ISA, segmentado, nueva tecnología
8051 original 12-cycle @12MHz, max 1 MIPS
8051 single-cycle @100MHz, max 100 MIPS

AT89LP51 @20MHz ($2.57)
Menor energía para la misma tarea
“less power consumption (up to 80% savings)”

FPGA
IP Core



100% de compatibilidad binaria (ISA)
Poco queda de la organización inicial (ORG)
Implementado con tecnología actual (TEC)



ADuC832 https://www.analog.com/en/products/aduc832.html