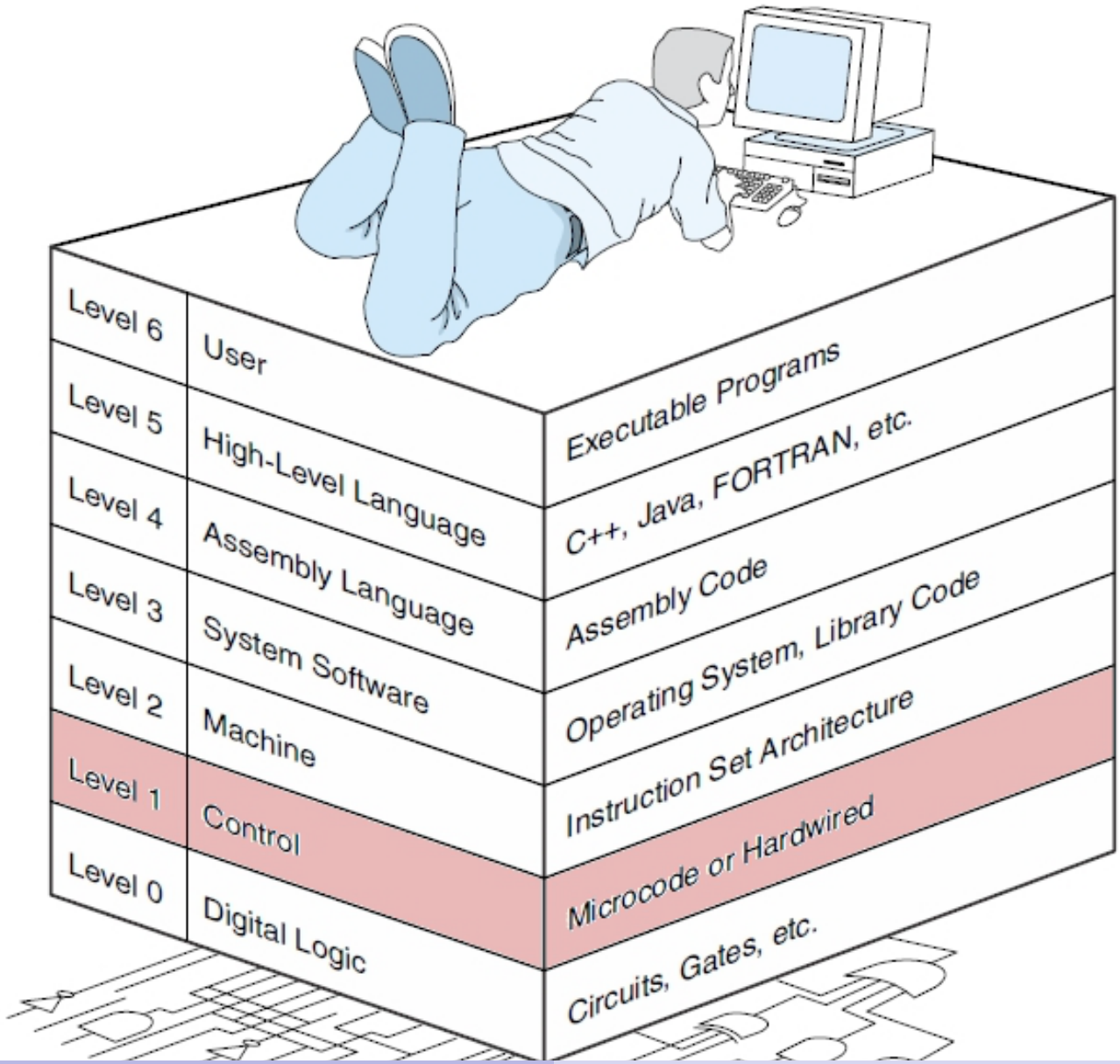
Microarquitectura

Esteban Mocskos[1]
basado en trabajo de Diego Garverbetsky[1], Marcelo Risk[1], Alejandro Furfaro[1],
Diego Fernández Slezak[1], Juan Pablo Galeotti[1],
Fernando Schapachnik[1]

[1]Departamento de Computación, FCEyN,
Universidad de Buenos Aires, Buenos Aires, Argentina

Organización del Computador I,
1er cuatrimestre de 2018

(2) ¿Dónde estamos?



(3) Zoom en el ciclo de instrucción

- Concentrémonos en el FETCH.
- ¿A dónde se trae la instrucción?
  - MAR (Memory Address Register): es un registro conectado a las líneas de dirección del bus que indica la dirección de la celda de memoria a leer.
  - MBR (Memory Buffer Register): otro registro, conectado a las líneas de datos del bus que indica el valor a escribir en la memoria, o el último leído.
  - PC
  - Instruction Register (IR): aquí va la última instrucción leída.

(4) FETCH en detalle

- Entonces, un FETCH consiste en:
  - t_1 MAR $\leftarrow$PC
  - t_2 MBR $\leftarrow$[MAR] (esto lo hace el bus, pero hay que esperar)
  - t_3 IR $\leftarrow$MBR
- ¿Y cómo es EXECUTE?

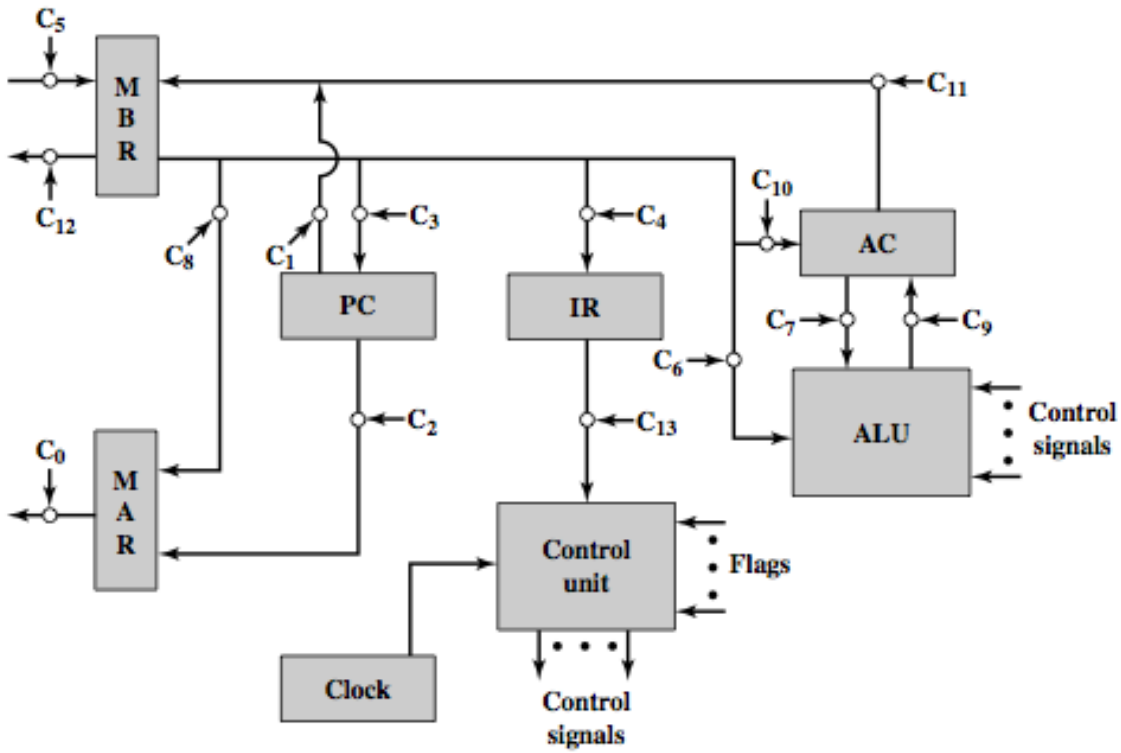
(5) Ejemplo de EXECUTE

- Analicemos `ADD R1, [DIR]`
  - t_1 MAR $\leftarrow$IR[op] (la parte de IR correspondiente al operando)
  - t_2 MBR $\leftarrow$[MAR] (esperar)
  - t_3 R1 $\leftarrow$R1 + MBR
- ¿Y si interviniese la ALU?

(6) Microinstrucciones

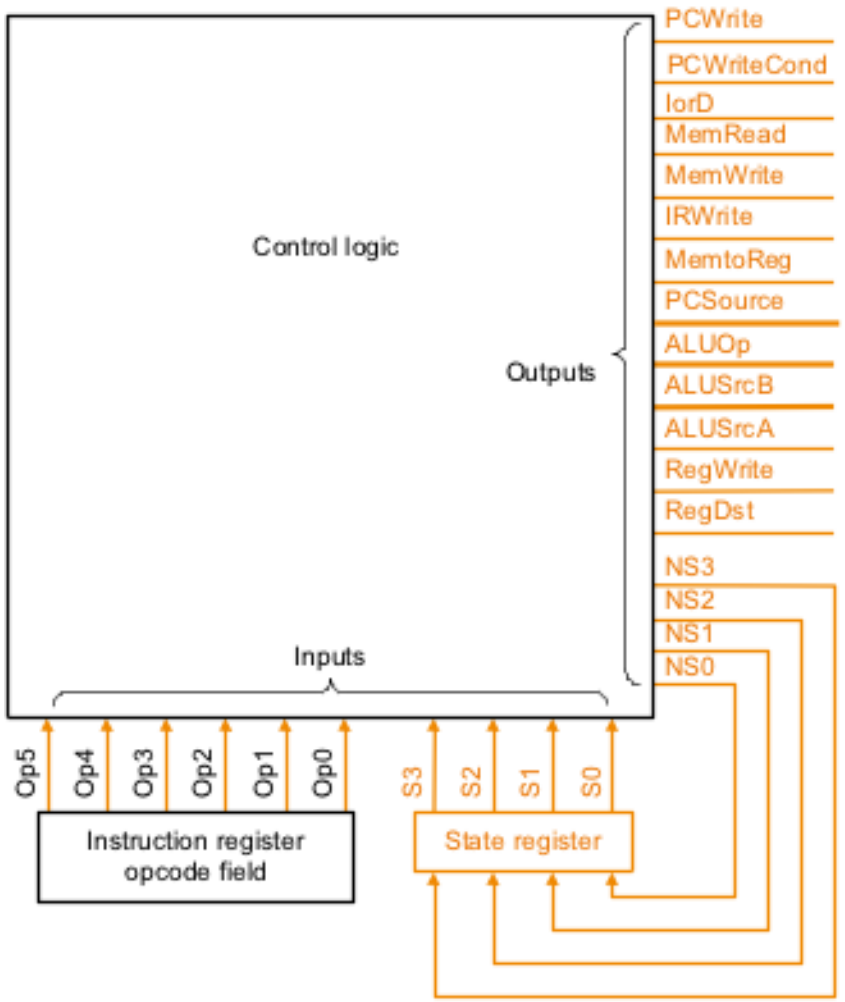
- Es decir, cada instrucción está compuesta por microinstrucciones.
- Al igual que las partes del ciclo de instrucción.
- Estas microinstrucciones realizan pasos muy simples.
- Poner cosas en registros, activar circuitos, prender/apagar líneas.
- Entonces, la UC tiene que ejecutar estas microinstrucciones.
- ¿Y cómo hace?
- Mediante señales de control.

(7) Señales de control



- Entonces, "MAR ←PC", es activar C_2.
- MBR ←memoria es activar C_5.
- "Memoria ←MBR" es activar C_{12}.
- ¿Cómo sabe la UC en que paso está?
- A través de la noción de estado.
- Ejemplo: 0000 $\rightarrow$ t_1 de FETCH, 0001 $\rightarrow$ t_2 de FETCH, ..., 1001 t_1 de EXECUTE, etc.

(8) Una UC con estados y señales



- En esta imagen las señales de control tienen nombres simbólicos.
- Y la UC retroalimenta su estado.

(9) La UC como una tabla de verdad

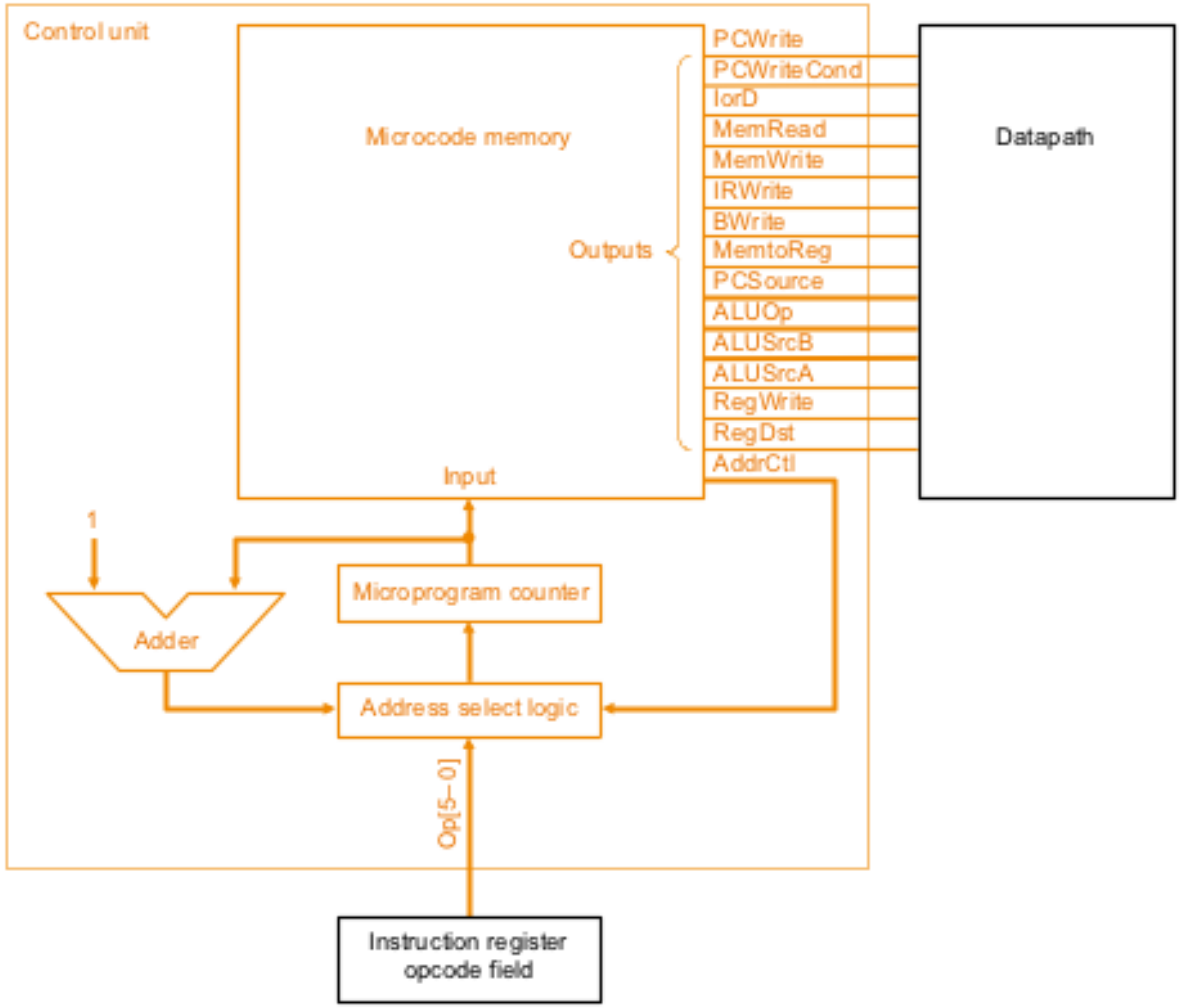
- Notemos que podríamos implementar a la UC como una tabla de verdad.
- Las columnas representan el estado, los inputs, el nuevo estado y los outputs.
- El input lo constituyen las señales de control y los registros internos que provienen entre otras cosas de decodificar la instrucción.
- ¿Qué pinta tiene esa tabla?
  - Supongamos 10 entradas y 20 señales de salida.
  - 2^{10} filas de al menos 20 bits c/u: $20 \times 2^{10} = 20$ Kbits
  - Muchas de ellas podrían estar repetidas.
  - Si no queremos usar esa tabla en ROM podemos usar un *Programmable Logic Array*.
- Esto se llama UC "cableada" (hardwired).

(10) Microcontrol cableado

- Una UC cableada es más rápida.
- Pero poco flexible.
- ¿Y si queremos hacer algún cambio?
- ¿O corregir algo?

(11) UC microprogramada

- Podríamos pensar en algo así:



(12) Microprogramación

- Una buena solución para microarquitecturas que deben soportar cientos de instrucciones, modos, etc.
- Las señales se especifican simbólicamente usando microinstrucciones.
- Se define el formato de la microinstrucción, estructurado en campos.
- Luego, cada campo se asocia a un conjunto de señales.

(13) Ejemplo de formato de microinstrucción

Campo	Función
ALU control	Qué debe hacer la ALU en este ciclo
SRC1	1er operando ALU
SRC2	2do operando ALU
Register Ctrl	Lectura/escritura de registros.
Memoria	Lectura/escritura de mem.
PC Write Ctrl	Escritura del PC.
Secuencia	Cómo elegir la próxima microinstrucción.

(14) Microprogramación

- A la hora de ejecutar una intrucción, una vez que está en IR:
  - Se toma el opcode como índice en una memoria de microinstrucciones.
  - Se carga esa dirección en un especie de "micro" PC.
  - Se ejecuta el microprograma.

(15) Control microprogramado

- Es más lento.
- Más caro (más barato en diseño).
- Mucho más flexible.
- Es lo que se usa en los procesadores Intel actuales.

(16) Bibliografía

- Stallings, capítulos 16 y 17.
- Null, capítulo 4.
- Tanembaum, capítulo 4.