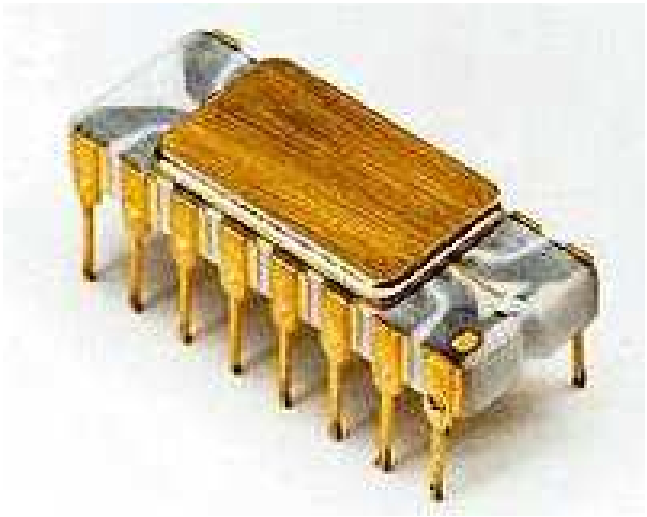
Construcción de Ufamaf:

Un Microprocesador Flexible en VHDL



Lic. Eduardo A. Sanchez

esanchez@famaf.unc.edu.ar

¿Porque estudiar la arquitectura de una computadora?

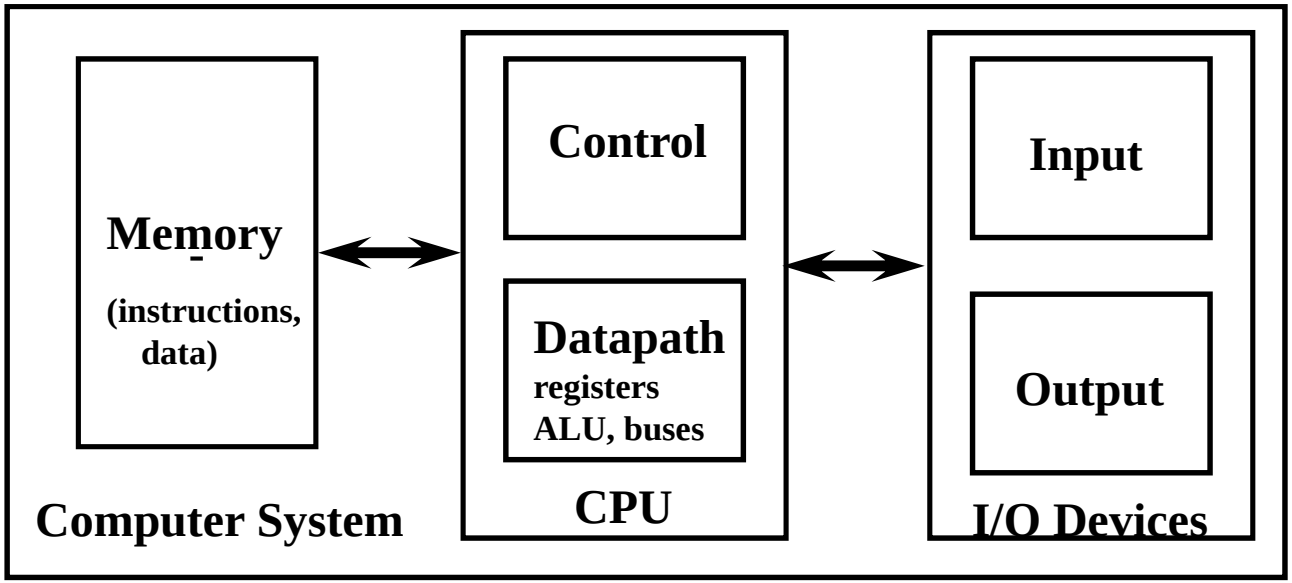
- Para poder construir una computadora.
- Para impresionar a sus familiares y amigos.
- Para entender la performance
  - Ciclos por instrucción
- Para entender los costos involucrados.
- Para entender los compiladores y otras herramientas.
- Porque trabajan en Intel o AMD. ☺

Introducción

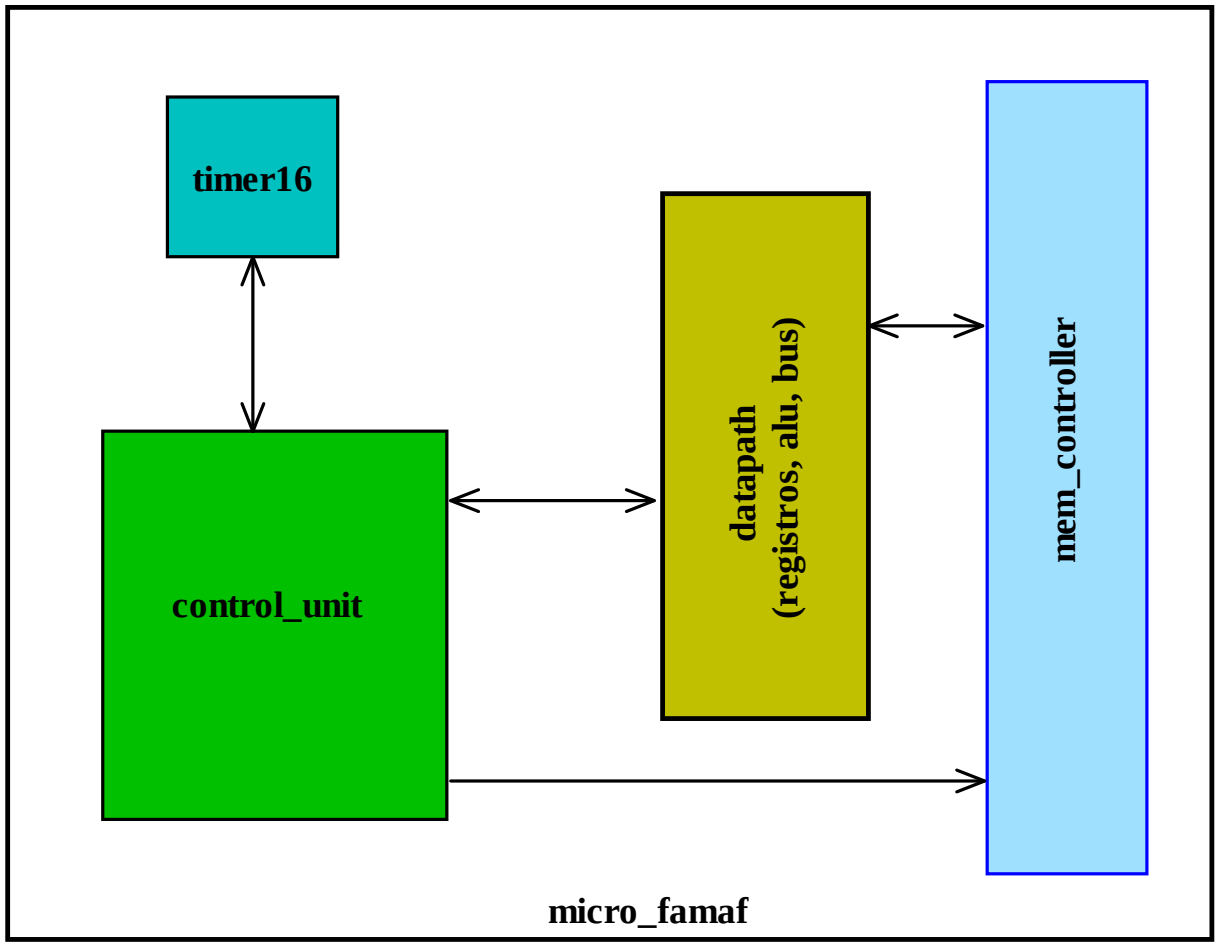
- Vamos a seguir los principios de un micro básico descripto por Morris Mano, con varias modificaciones.
- Veremos los componentes necesarios, sus interconexiones y el funcionamiento en general.
- Pequeños módulos en VHDL crean algo poderoso como un procesador flexible.

Base en modelo Von Neumann

- **Unidad Central de Proceso (CPU):** Unidad de Control (decodificación de instrucción, secuenciación de las operaciones, etc), Datapath (registros, ALU, Buses).
- **Memoria**: Tanto para almacenar instrucciones como datos.
- **Sub-sistema de Input/Output (I/O):** I/O bus, interfaces, devices.



El modelo de Ufamaf

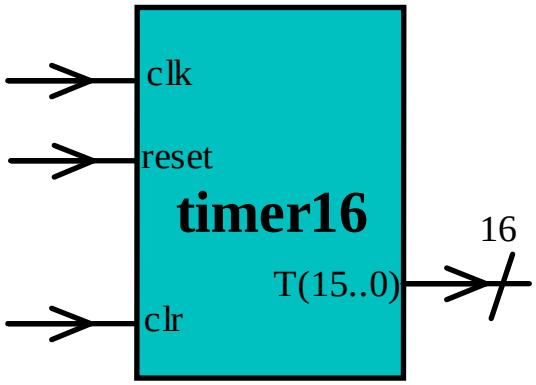


Es un procesador de 16 bits.

Timer

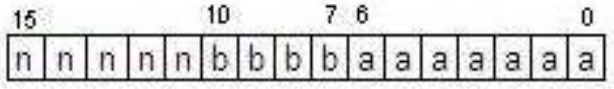
- Necesitamos un generador de secuencias:

  T0, T1, .. T15:
  - Solo hay un T activado por vez.
  - En cada pulso, se incrementa.
  - Tiene clear síncrono que vuelve a T0
  - T0 = T(0,..,0,1)
  - T1 = T(0,..,1,0), y así sucesivamente..



Memoria

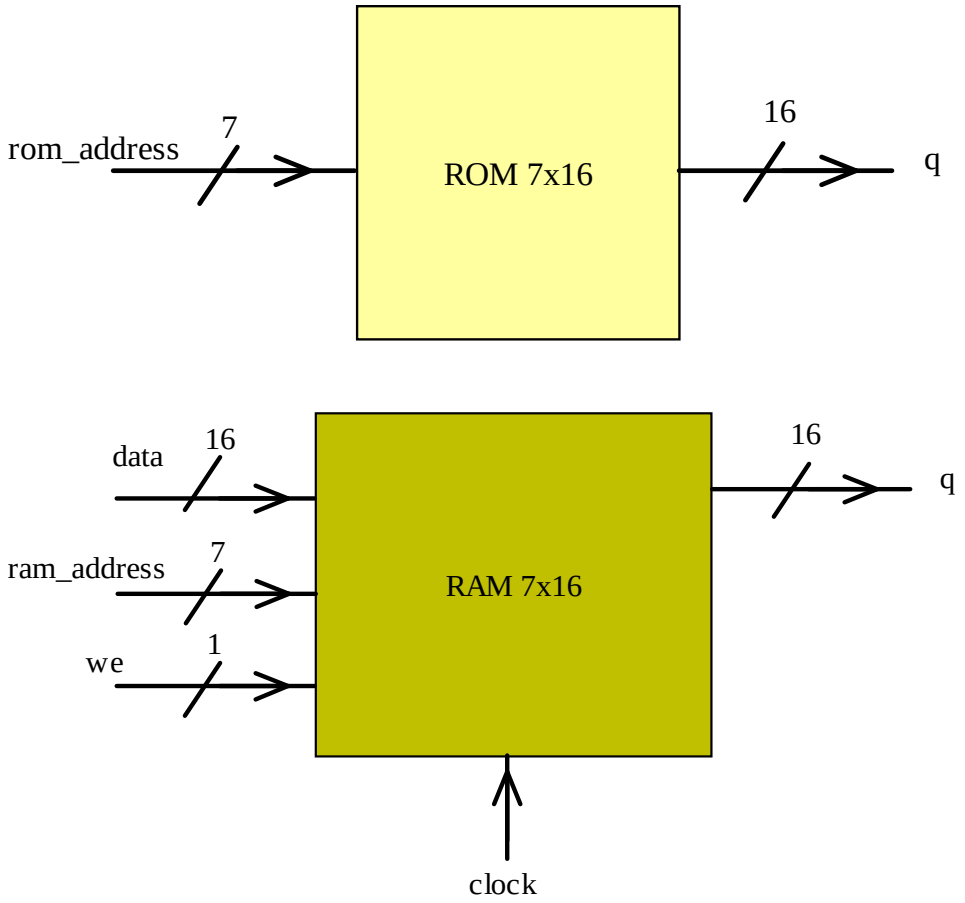
- Es capaz de direccionar **solo** 2k de memoria. (necesitamos 11 bits, y tenemos 16..)
- Esta dividida en bancos (RAM y ROM) de 128 palabras de 16 bits c/u (del bit 10 al 7 de la dirección, seleccionan el banco)
- Formato de dirección:



n: no importa, b: banco, a:dirección dentro del banco

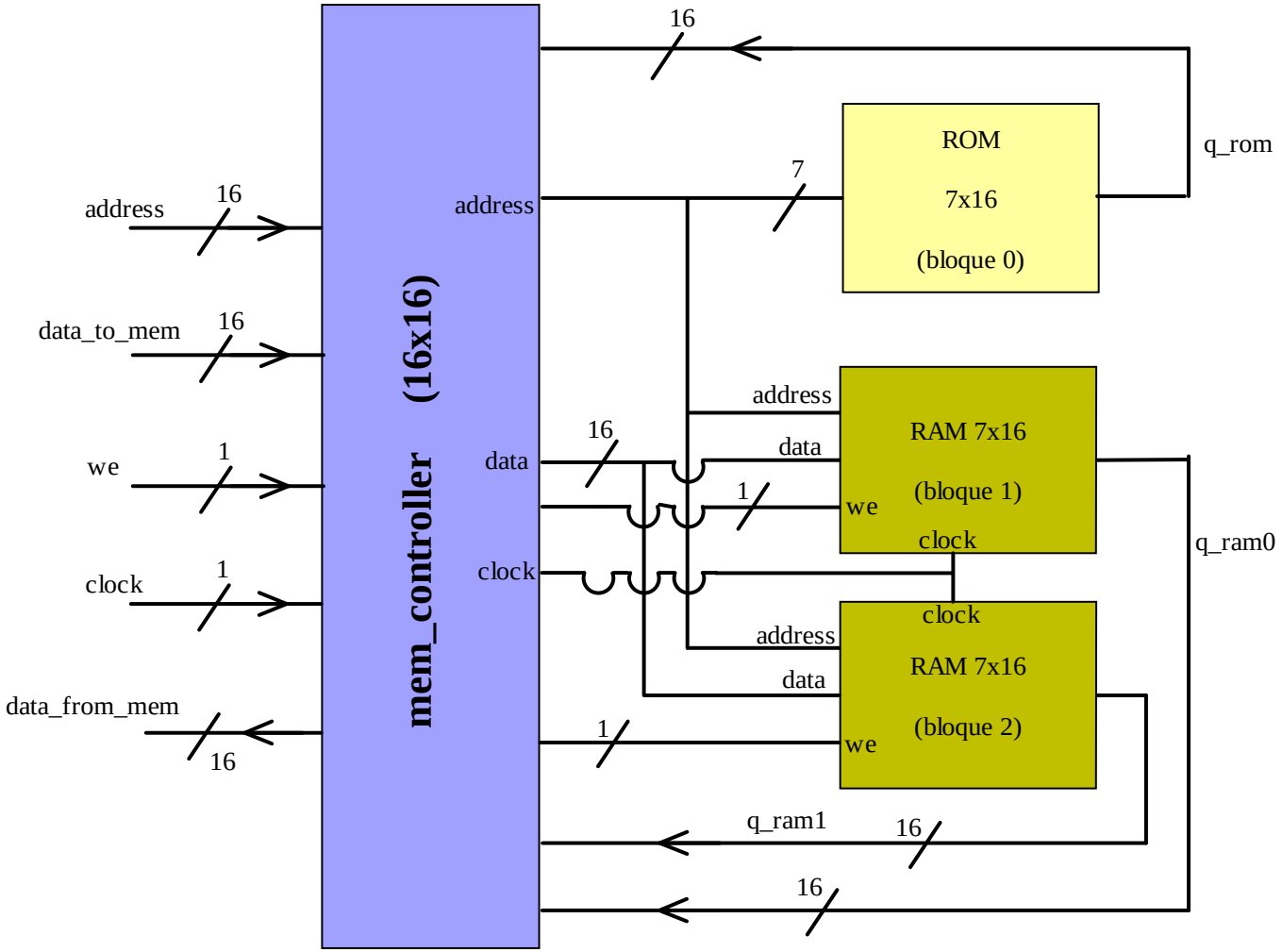
Memoria

- Configuración inicial de tres bancos:
  - Banco 0 = ROM
  - Banco 1 = RAM
  - Banco 2 = RAM



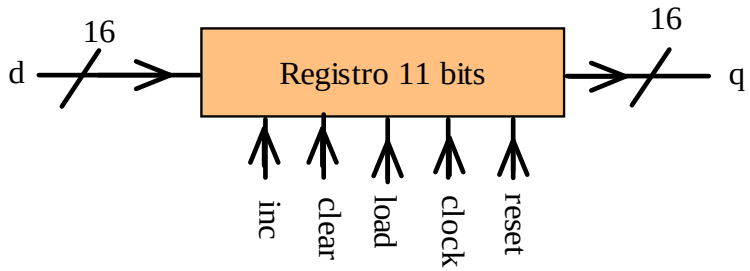
Memoria

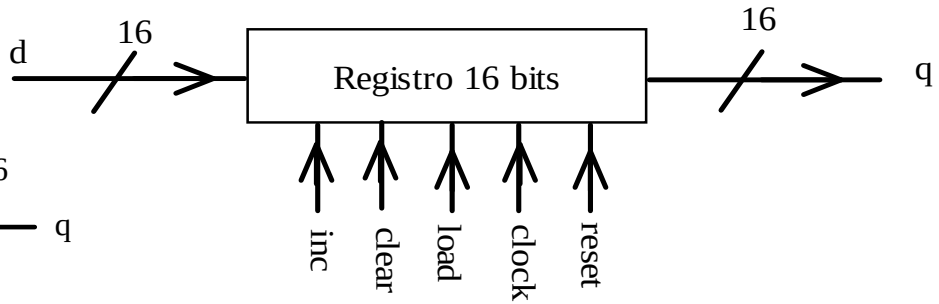
- Controlador de memoria:



Datapath: Registros

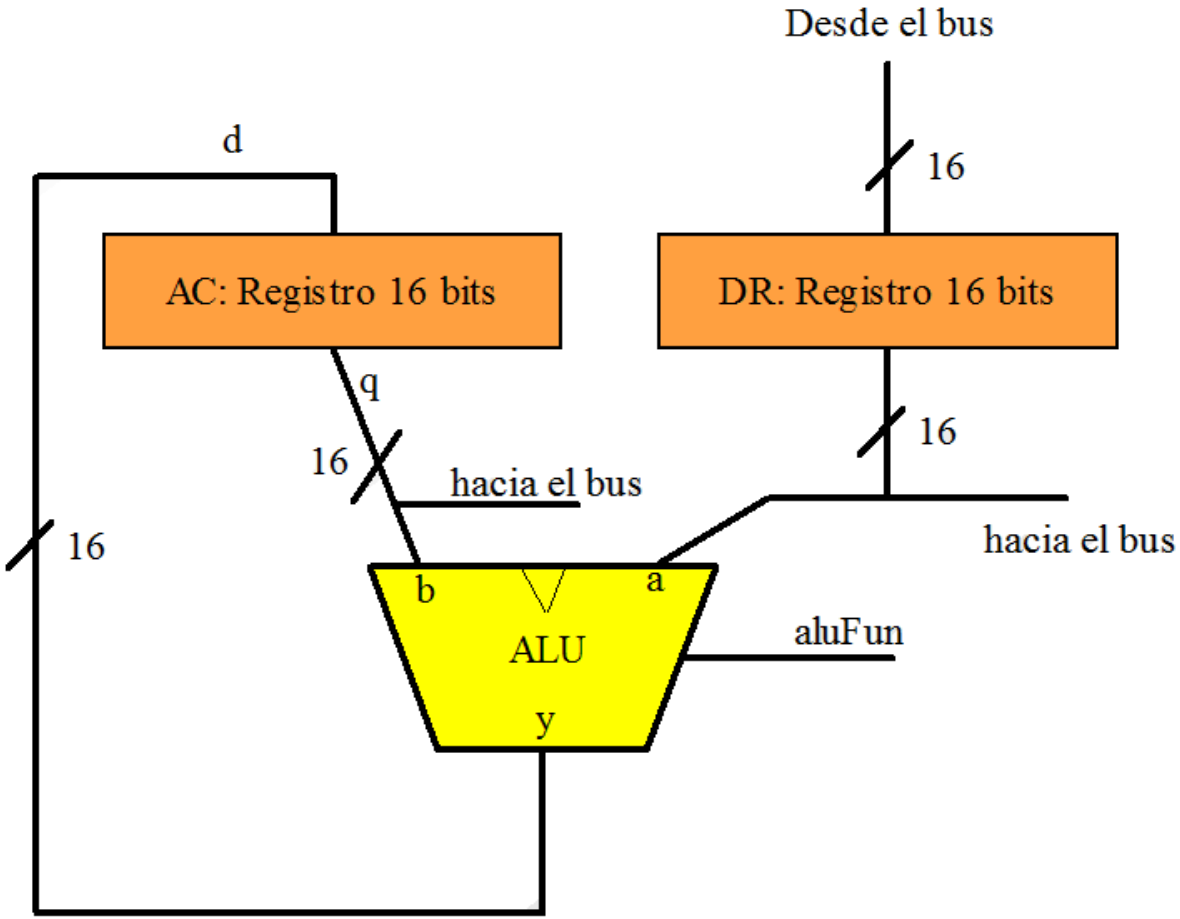
- Posee 5 registros de 16 bits:
  - AR*: “Address Register”: contiene la dirección del operando.
  - PC*: “Program Counter”: contiene la dirección de la instrucción a ejecutar.
  - IR: “Instruction Register”: contiene la instrucción a ser/siendo ejecutada.
  - DR: “Data Register”: se utiliza como registro intermedio entre la memoria y la ALU.
  - AC: Registro acumulador.





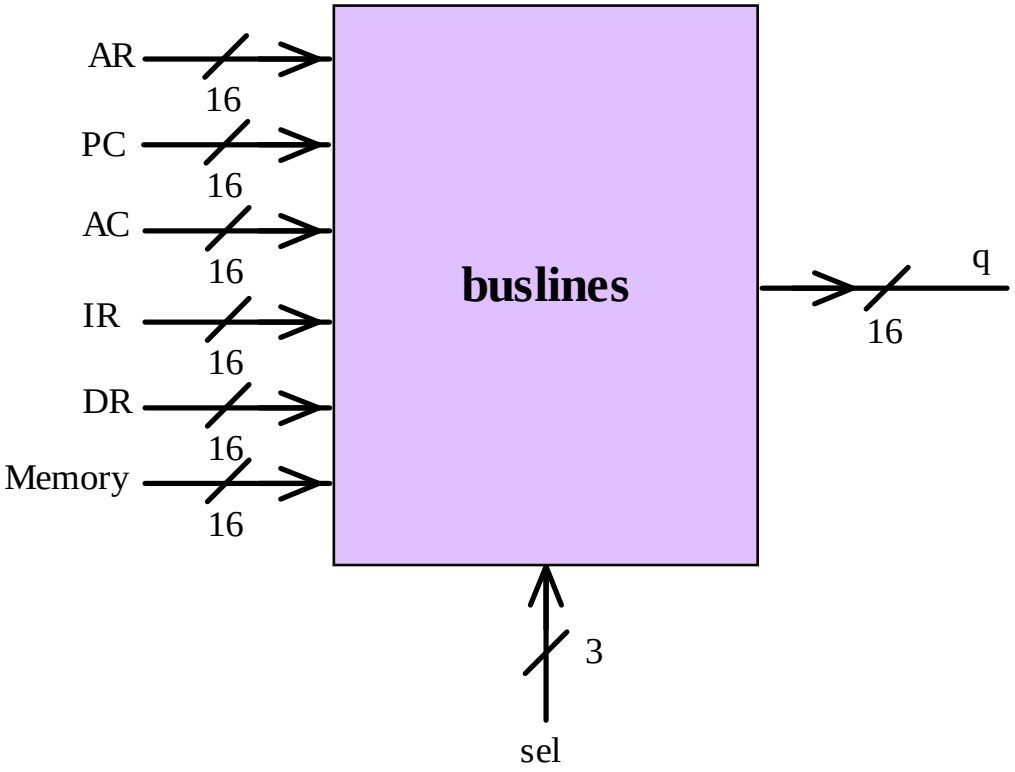
*: En realidad son registros de 11 bits, pero d y q son de 16

Datapath: ALU



aluFun	y
0000	a
0001	a+1
0010	b-1
0011	b
0100	b+1
0101	b-a
0110	a+b
0111	a+b+cin
1000	Not a
1001	Not b
……	……

Datapath: Bus de conexión

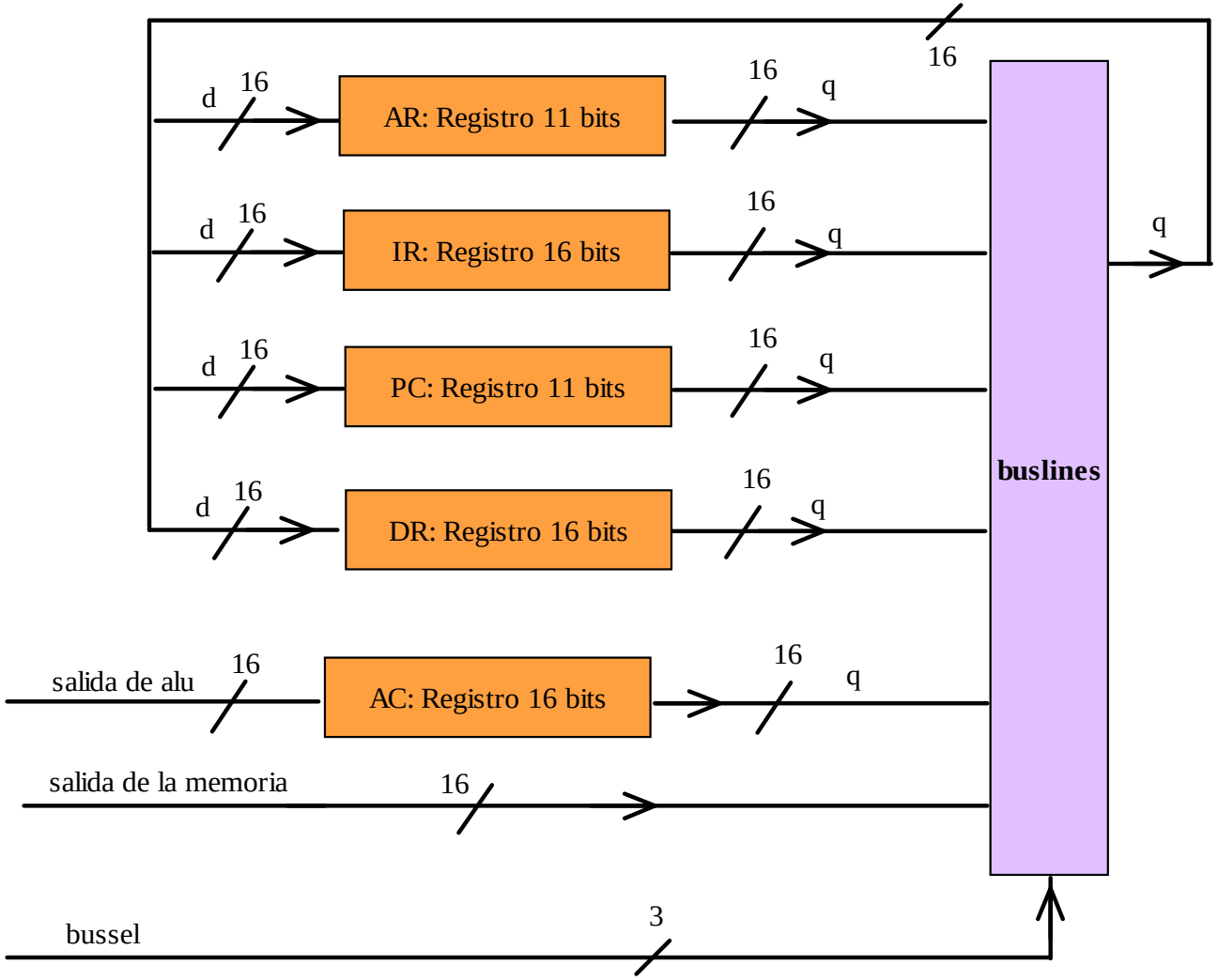


sel	Q
001	AR
010	PC
011	DR
100	AC
101	IR
111	Memory
resto	1111111111111111

Datapath: Bus de conexión

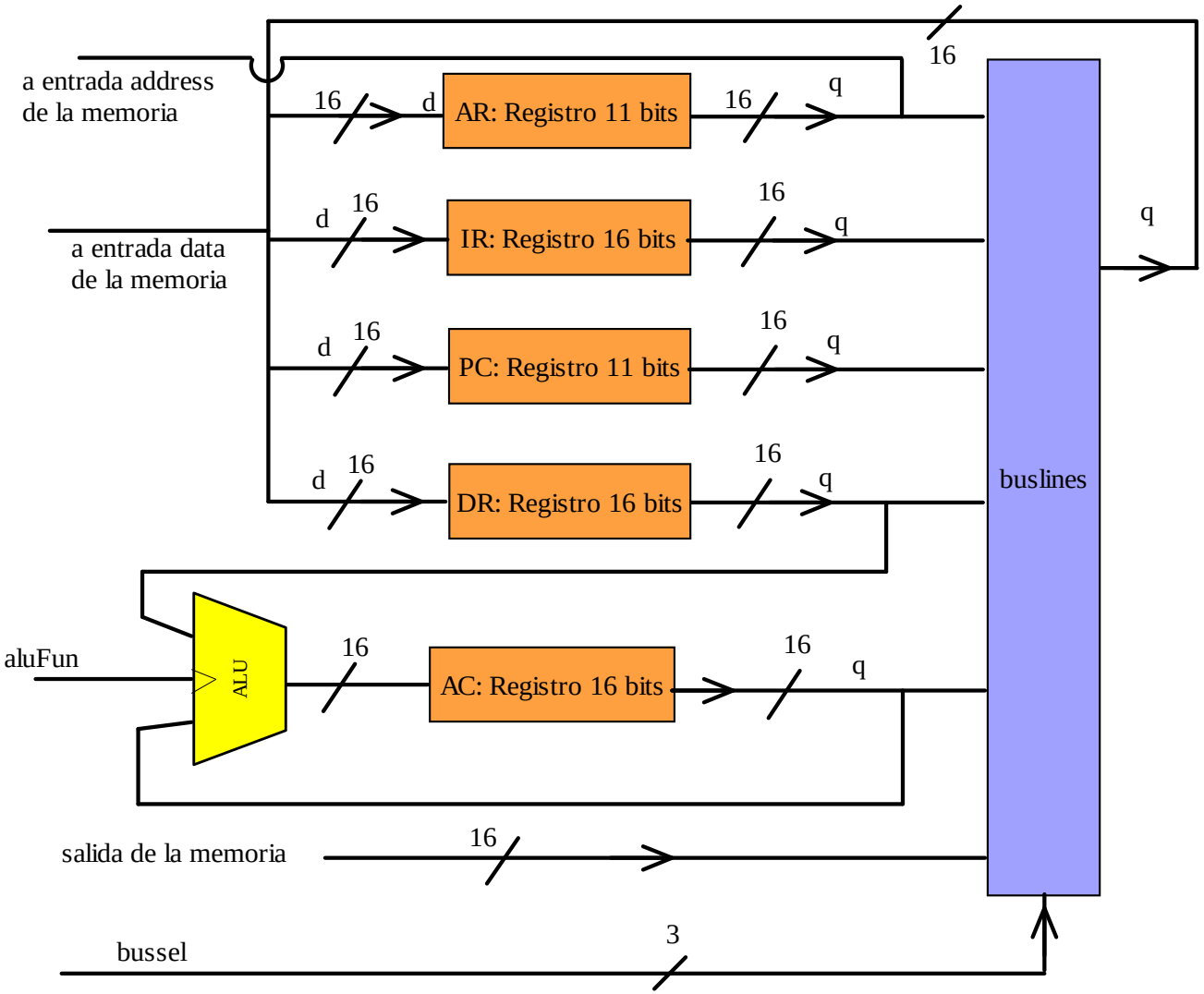
- Es un multiplexor simple.
- Sus entradas son las salidas de los registros y la memoria.
- Su salida, realimenta las entradas de los registros.

Datapath: Bus de conexión

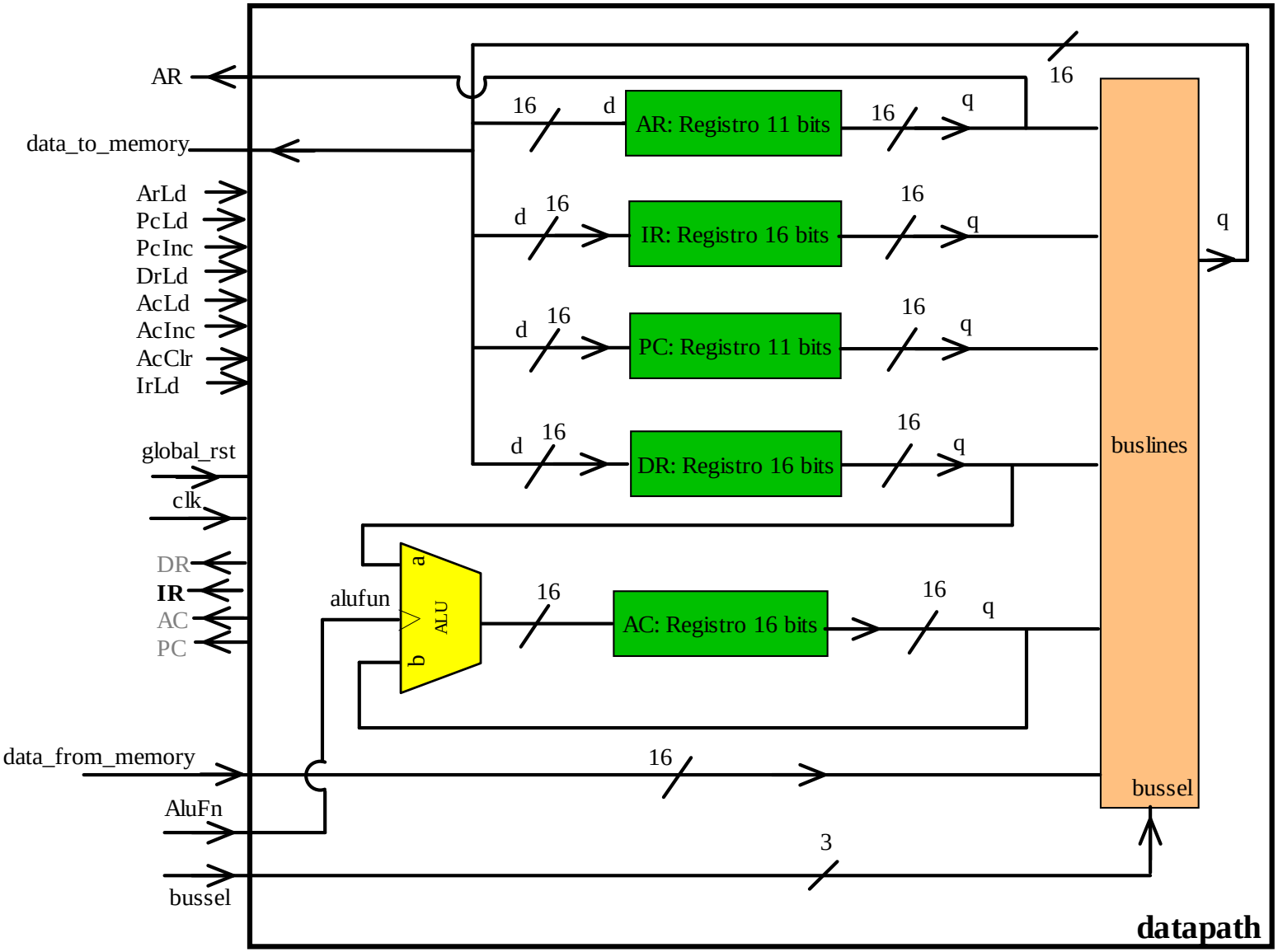


bussel	Q
001	AR
010	PC
011	DR
100	AC
101	IR
111	Memory
resto	1111111111111111

Datapath: Uniendo piezas

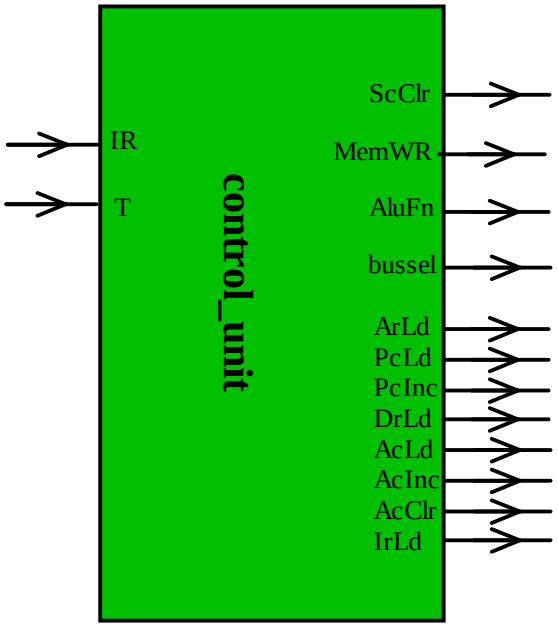


Datapath: Uniendo piezas



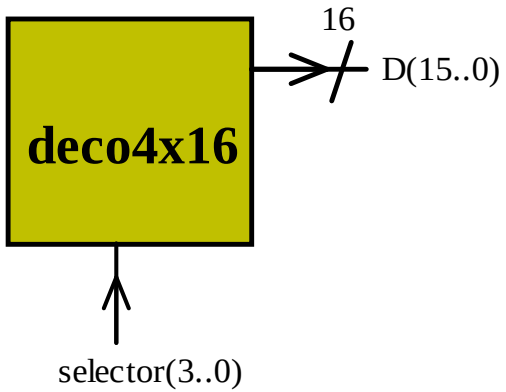
Lógica de Control

- Es el encargado de orquestar todas las señales de la micro-arquitectura, para poder ejecutar la instrucción.
- Puramente combinacional.

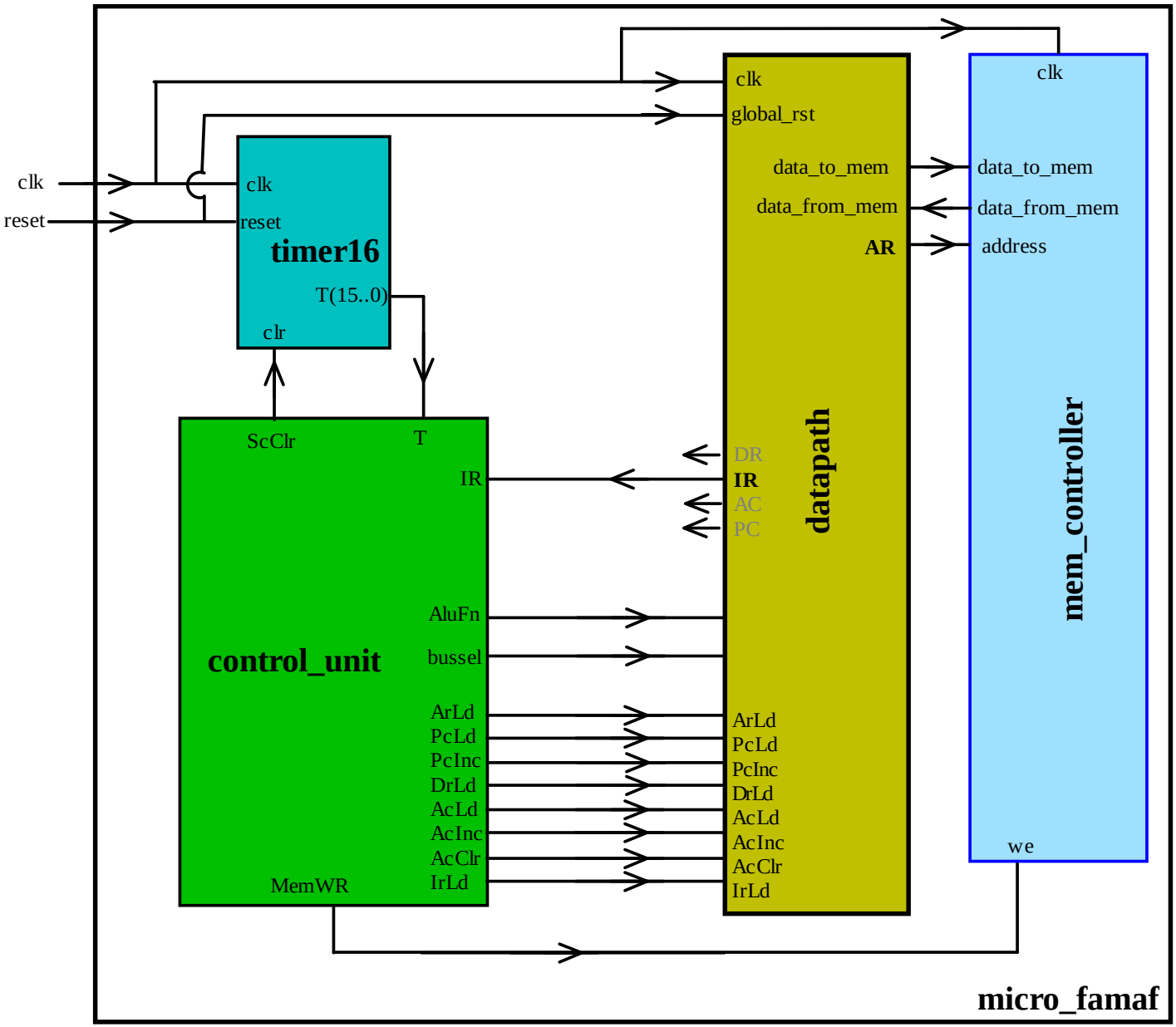


Lógica de Control: Decodificador

- Vamos a necesitar un decodificador 4x16, para facilitar la lógica en la unidad de control.
- La idea es que el “opcode” de la instrucción, habilite una sola salida “D” (D0..D15) y que podamos operar directamente sobre D.
- Ej. “D7 and T5”, “D1 and T3”, etc..



Micro Famaf Completo



Preguntas?

esanchez@famaf.unc.edu.ar