


Microprocesadores del pasado: paleosaurios y pirañas


José L. Briz

Prof. Titular, Depto. de Informática e Ing. de Sistemas

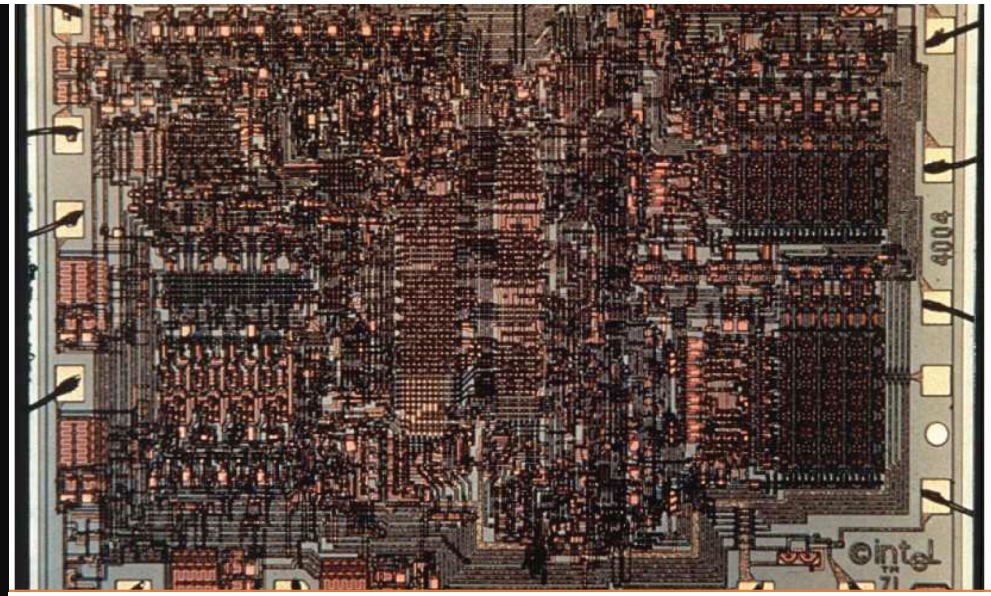
Grupo de Arquitectura de Computadoras GaZ

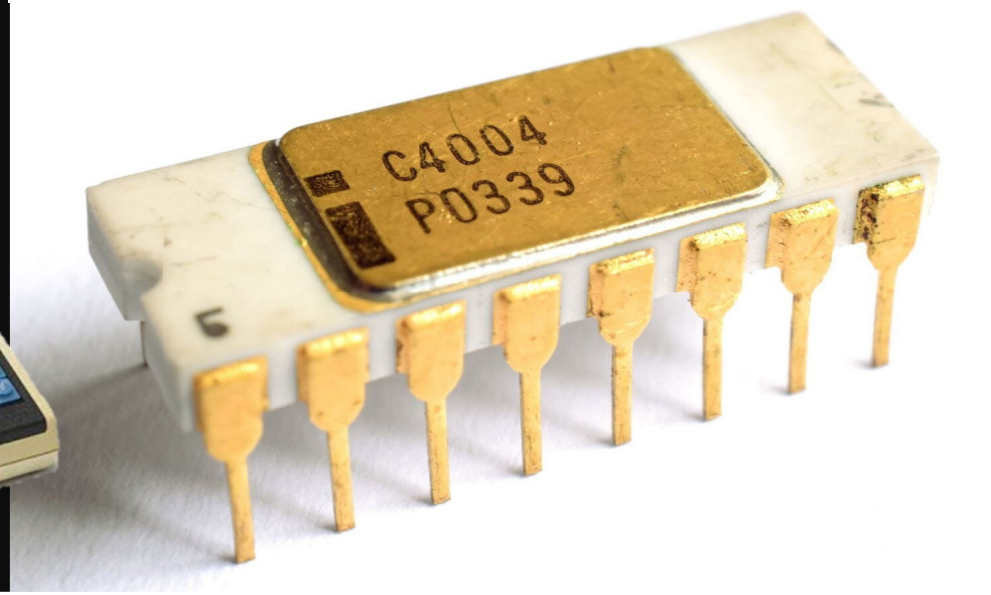
Instituto de Investigación en Ingeniería de Aragón (I3A)

UNIVERSIDAD DE ZARAGOZA


Intel 4004 (1971/’74)

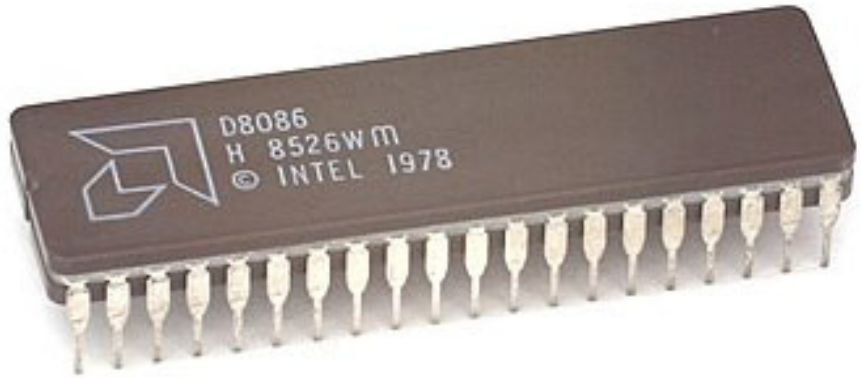
- Nodo 10 µ
- 2300 transistors
- 740 - 750 KHz (~1333 ns)
- 50 Kips
- Datapath 4 bits
- @ 12 bits (muxed)
- Aritmética BCD
- ~60 € de 2024?

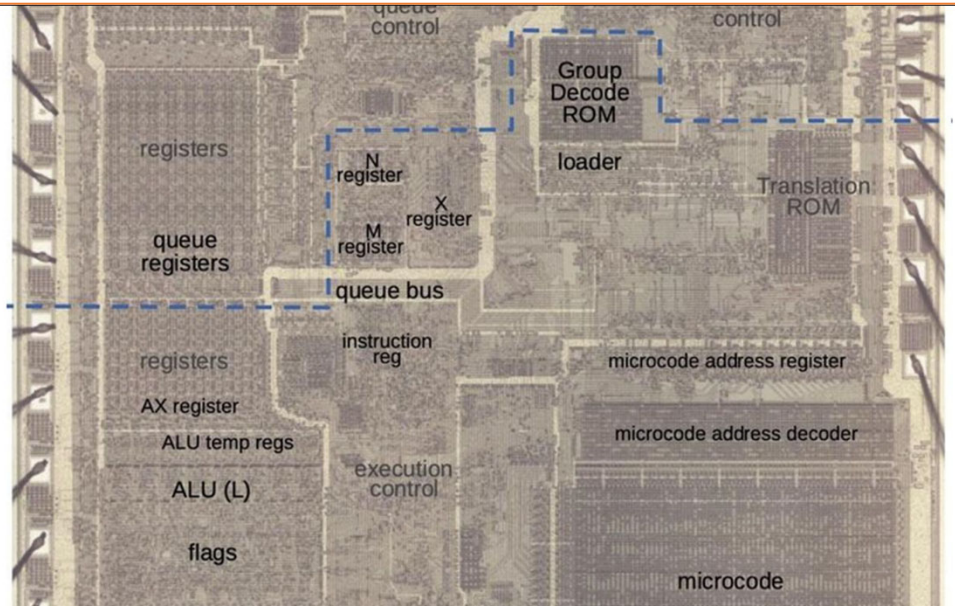




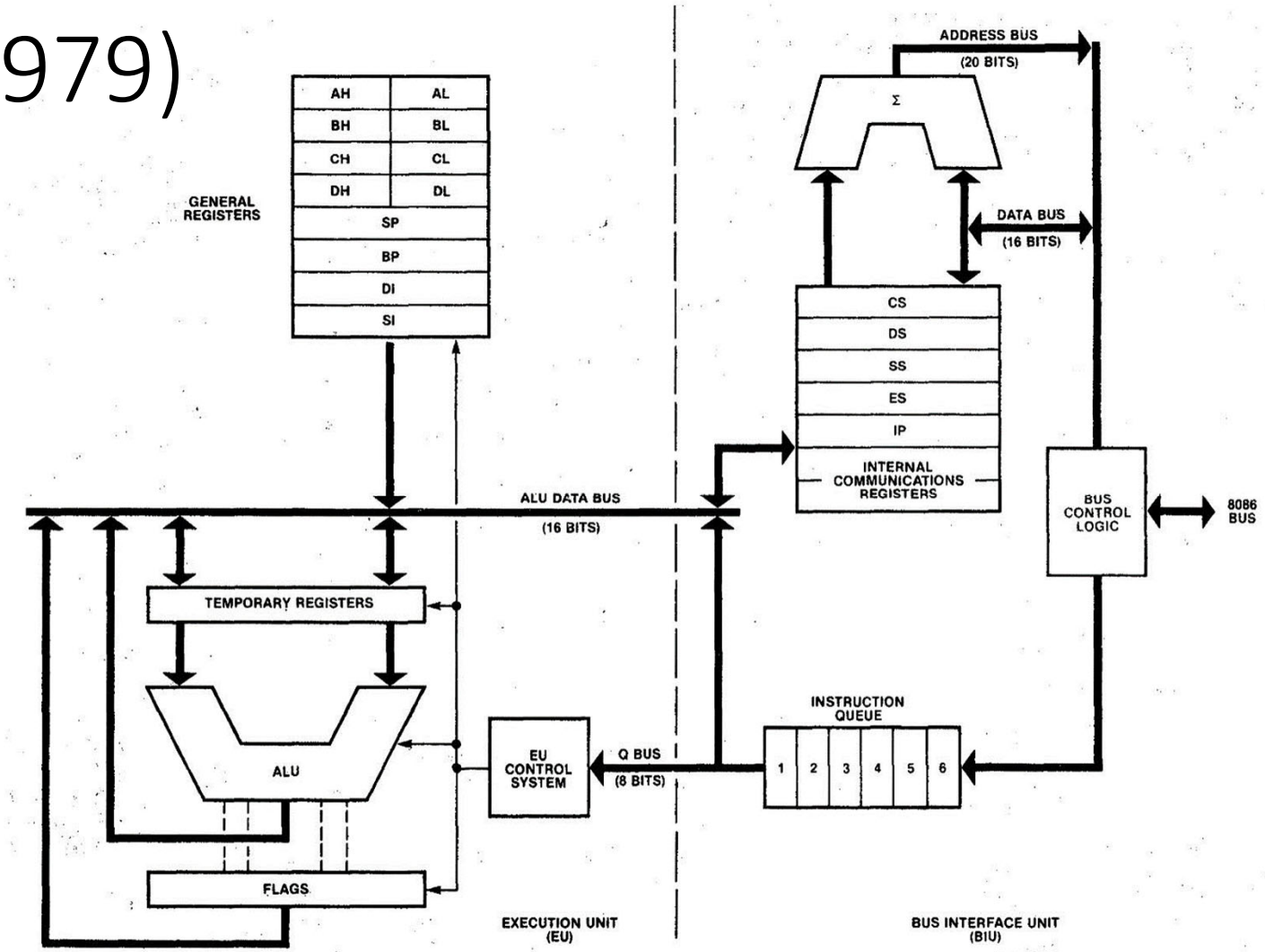
8086 (1979)

- 29K transistores (3µ)
- 5 a 10 (12) MHz
- Segmentación muy básica a tres niveles
  - BIU – EU
  - Microcódigo
  - Accesos a memoria



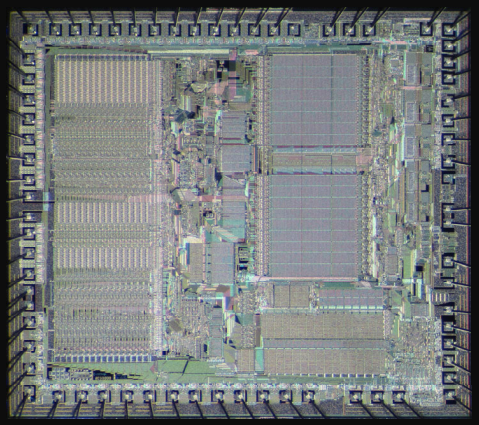


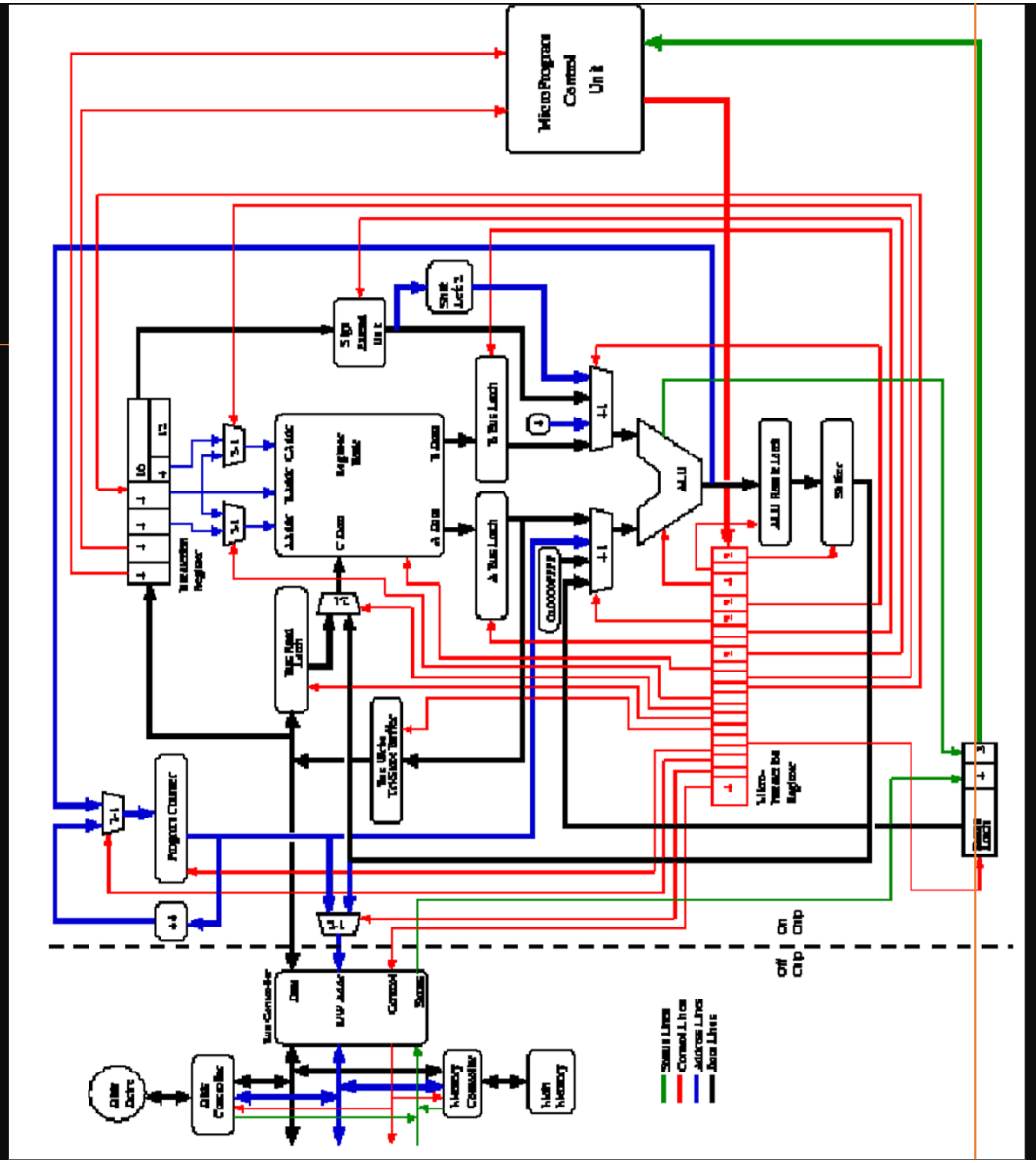
8086 (1979)


ADDRESS BUS
(20 BITS)
Σ
DATA BUS
(16 BITS)
GENERAL REGISTERS
AH
AL
BH
BL
CH
CL
DH
DL
SP
BP
DI
SI
CS
DS
SS
ES
IP
INTERNAL COMMUNICATIONS REGISTERS
BUS CONTROL LOGIC
8086 BUS
ALU DATA BUS
(16 BITS)
TEMPORARY REGISTERS
ALU
FLAGS
EU CONTROL SYSTEM
Q BUS
(8 BITS)
INSTRUCTION QUEUE
1
2
3
4
5
6
EXECUTION UNIT (EU)
BUS INTERFACE UNIT (BIU)

Motorola 68000

- Nodo 3- 5 µm (my guess!)
- 68000 transistores ☺
- 4 MHz to 16.67 MHz
- 32 b (datos, @)

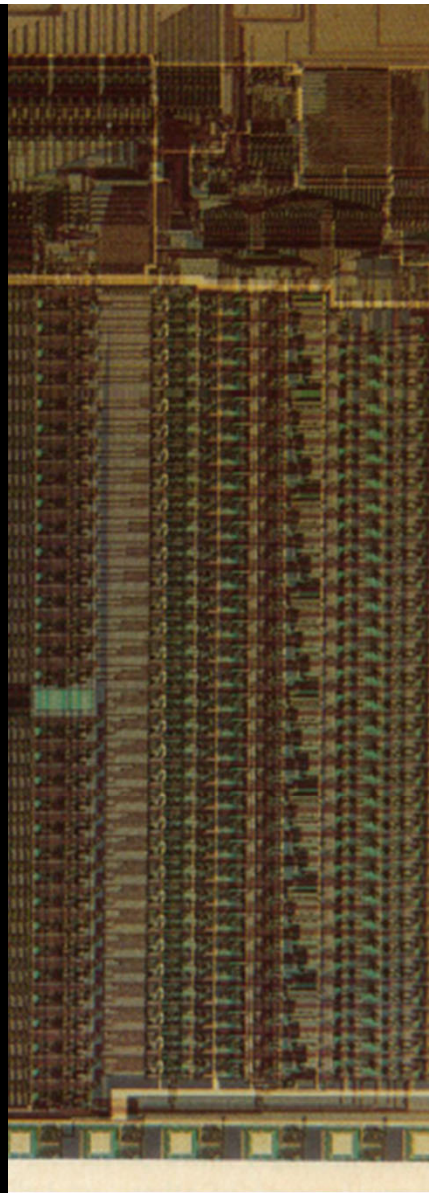




RISC I - 1982

- Patterson and Sequin, "A VLSI RISC," in Computer, vol. 15, no. 9, pp. 8-21, Sept. 1982

- 44.4200 transistors
- 32 instrucciones



APPLICATIONS

Computers

‘Superpower’ computers

The omnipresent microprocessor can hardly supplant the large high-speed computer in modeling complex systems and phenomena

- R. Sugarman, "Computers: ‘Superpower’ computers: *The omnipresent microprocessor can hardly supplant the large high-speed computer in modeling complex systems and phenomena*," in IEEE Spectrum, vol. 17, no. 4, pp. 28-34, April **1980**

Cray 1

12.5 MHz (160 Mflops!!!!)
5 Tm
115 Kw (con 8 MB)
8 (logic) + 16 (mem) frames
Max. mem. 8 MB
Regs. de 64 b
8 regs. Vect. x64b
8 regs. Escalares x 64b
64 regs backup 64b

8 regs @ 24 b
64 regs @ backup 24b

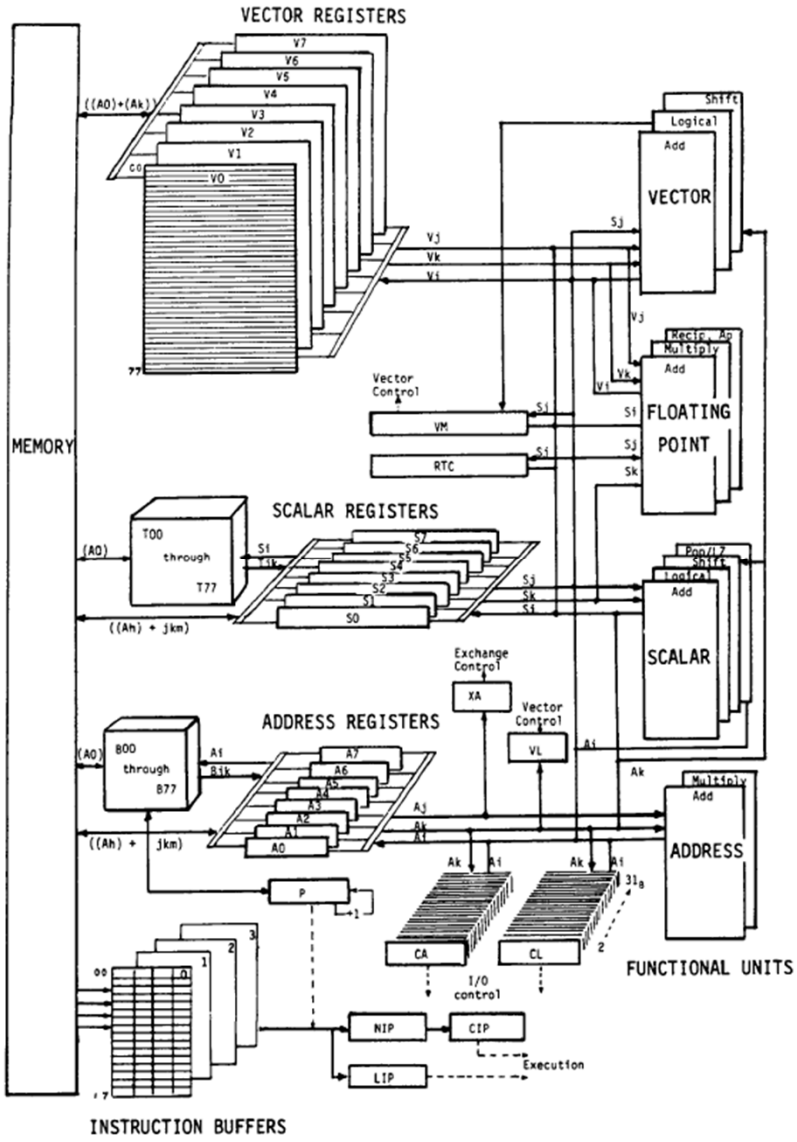
PC 22b

3 ALUs Vectoriales
4 ALUs escalares
3 fpALUs (shared)
Mc de instrucciones

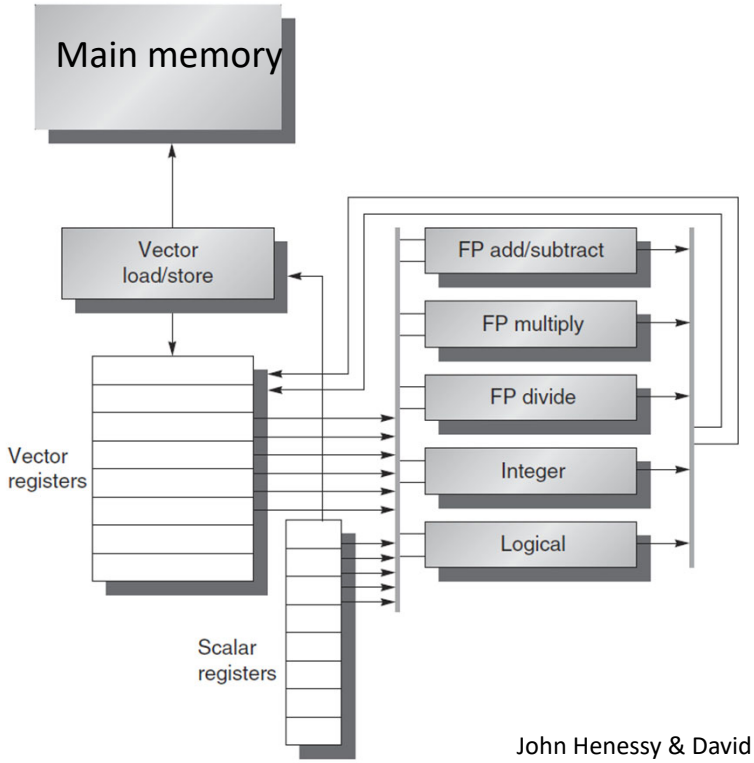
Sólo 4 tipos de chips

Richard M. Russell. 1978. The CRAY-1 computer system. Commun. ACM 21, 1 (Jan. 1978), 63–72

Fig. 5. Block diagram of registers.



Vector architectures



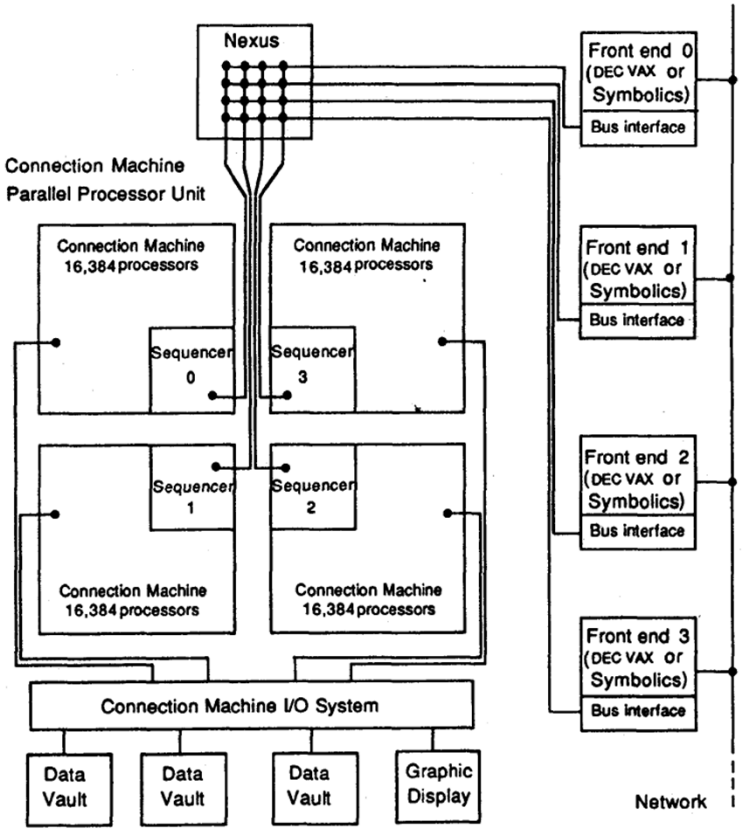

```
        L.D     F0,a        ;load scalar a
        DADDIU  R4,Rx,#512  ;last address to load
Loop:   L.D     F2,0(Rx)    ;load X[i]
        MUL.D   F2,F2,F0    ;a × X[i]
        L.D     F4,0(Ry)    ;load Y[i]
        ADD.D   F4,F4,F2    ;a × X[i] + Y[i]
        S.D     F4,9(Ry)    ;store into Y[i]
        DADDIU  Rx,Rx,#8    ;increment index to X
        DADDIU  Ry,Ry,#8    ;increment index to Y
        DSUBU   R20,R4,Rx   ;compute bound
        BNEZ    R20,Loop    ;check if done
```

Here is the VMIPS code for DAXPY.

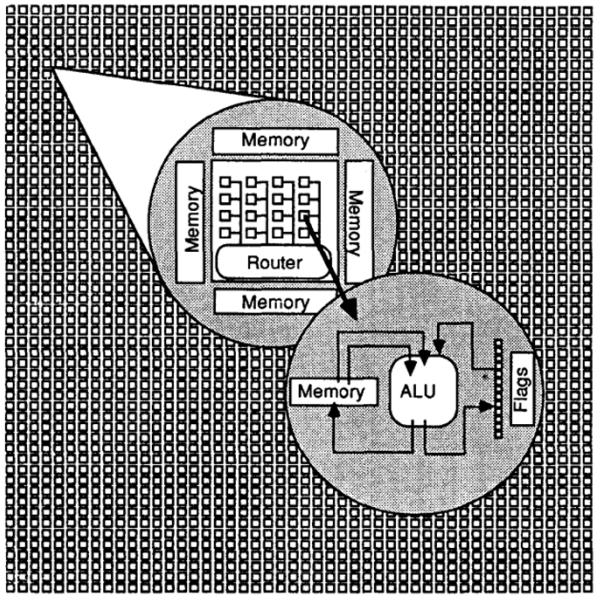
```
        L.D     F0,a        ;load scalar a
        LV      V1,Rx       ;load vector X
        MULVS.D V2,V1,F0    ;vector-scalar multiply
        LV      V3,Ry       ;load vector Y
        ADDVV.D V4,V2,V3    ;add
        SV      V4,Ry       ;store the result
```

John Henessy & David Patterson,
Computer Architecture A
Quantitative Approach 5 ed Fig. 4.2

Connection Machine (CM-1)



Idea: Tesis MIT 1980
Comercial: 1986



L. W. Tucker and G. G. Robertson, "Architecture and applications of the Connection Machine," in Computer, vol. 21, no. 8, pp. 26-38, Aug. 1988,

Convex (1986 / 1990)

- 50 Hz por CPU vectorial
- 100 MFLOPs
- 128 MB DRAM
- 16 GB HD
- ECL salvo el vector processor(CMOS gate arrays)

Facts - 1993

- Baskett, Forest, and John L. Hennessy. "*Microprocessors: From Desktops to Supercomputers*." **Science** 261, no. 5123 (**1993**): 864–71.

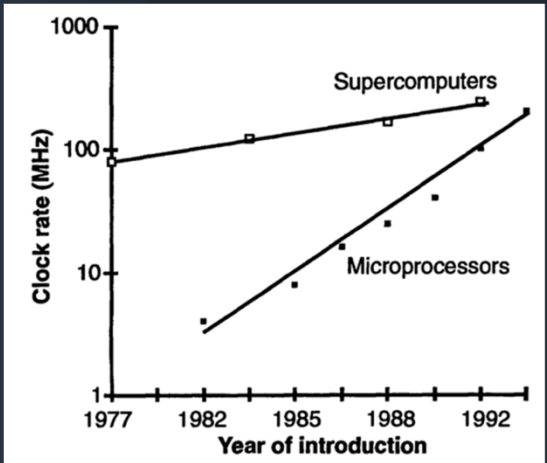
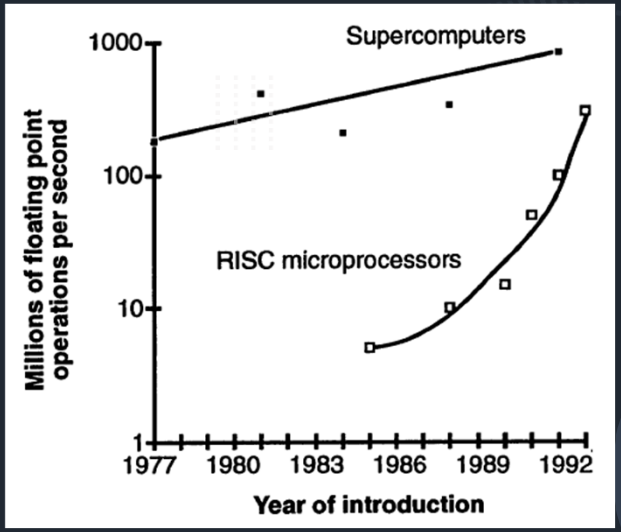
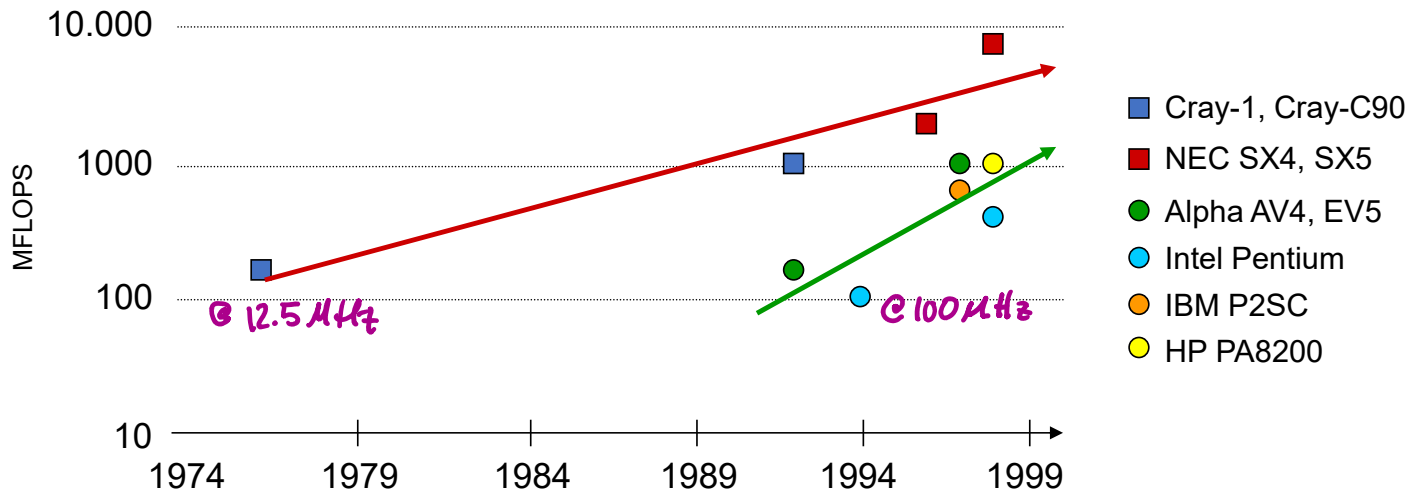


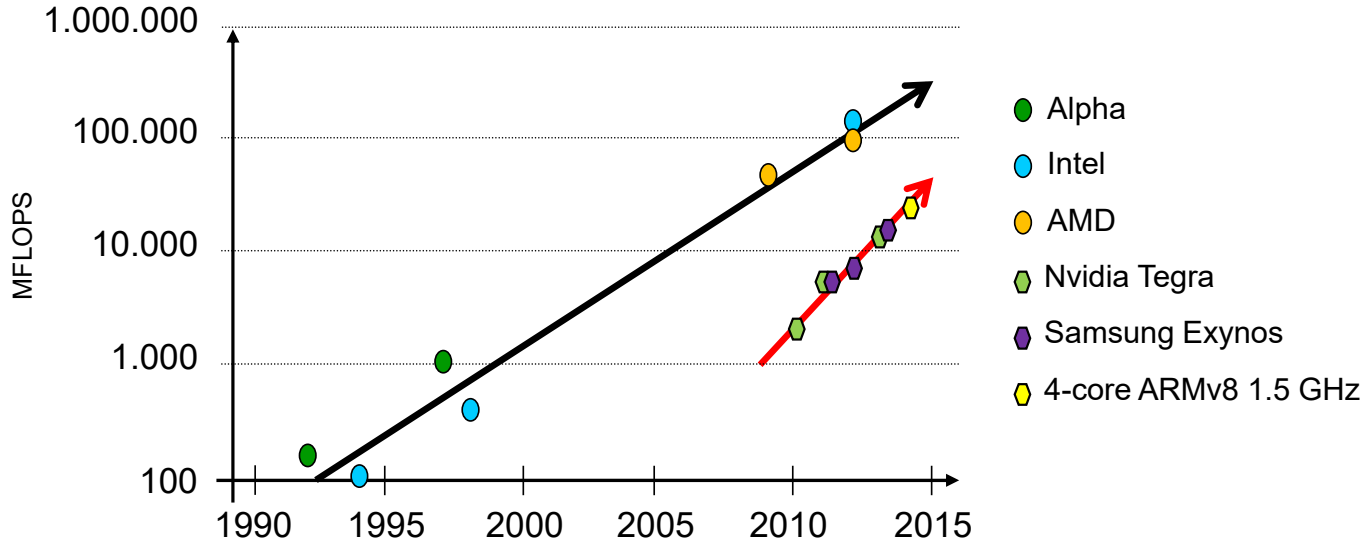
Fig. 3. Improvement in clock rate over time for Cray supercomputers and high-performance multiprocessors.





M. Valero. "Vector Architectures: Past, Present and Future". Keynote talk. ICS-11. IEEE-ACM. Melbourne, **1998**

[*Fandom-Micro - Spoiler*]



La Supercomputación: motor para la ciencia y la ingeniería

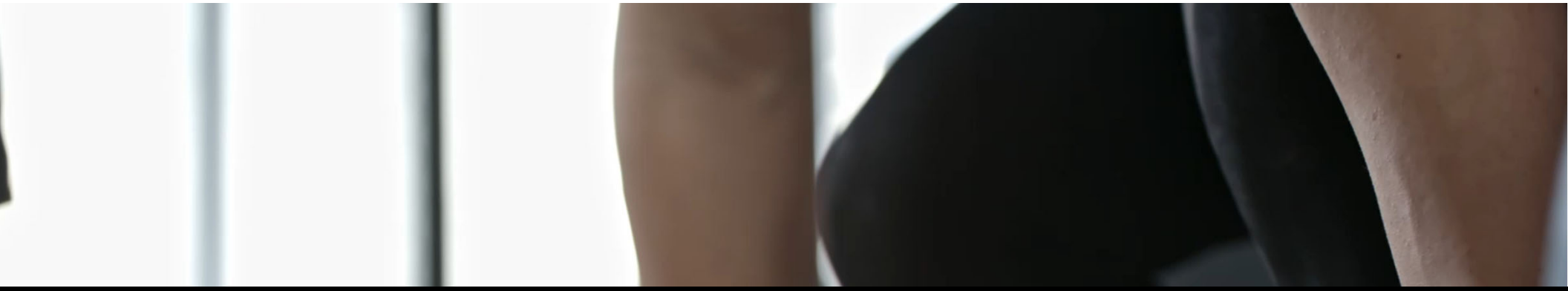
Mateo Valero, Director BSC

Ateneo EINA, 20 de marzo 2019

Connection Machine (CM-1) 1980 idea ->’86

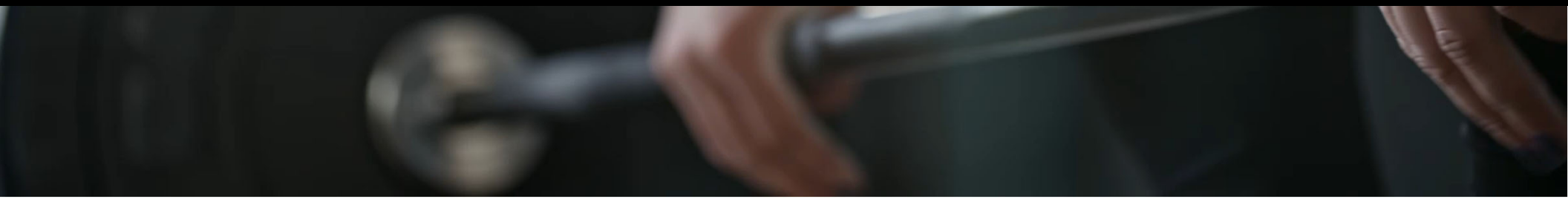
Thinking Machines Connection Machine models [hide]											
	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994
	Custom architecture								RISC-based (SPARC)		
Entry	—				CM-2a				—		
Mainstream	—	CM-1		CM-2			—		CM-5	CM-5E	
Hi-end	—				CM-200						
expansions											
Storage	—			DataVault					—		

Wikipedia, *Connection Machine*



$$T_{ex} = I \times CPI \times Tc$$

[pizarreo 1]



Tc

- Moore’s Law
- Dennard’s Scaling
- Diseño

Escalado del transistor = integración de funciones

1959. First commercial planar transistor, Fairchild Semiconductor

1964. MOSFET, the foundation for all future transistor technology, General Microelectronics

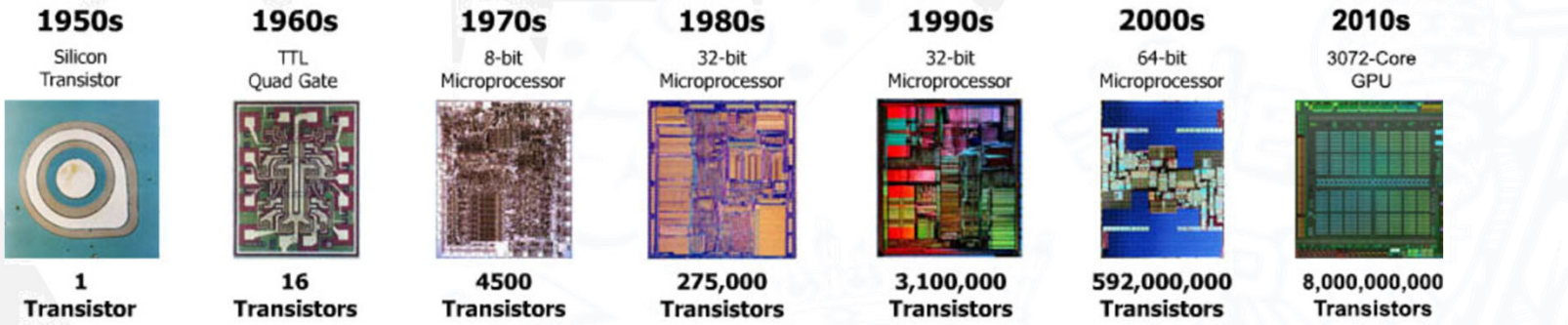
1971. Silicon gate technology, Intel 4004 (10 µm) and 8080 (10 µm)

1977-78. High-density, short-channel MOS, Intel 8086 (3 µm)

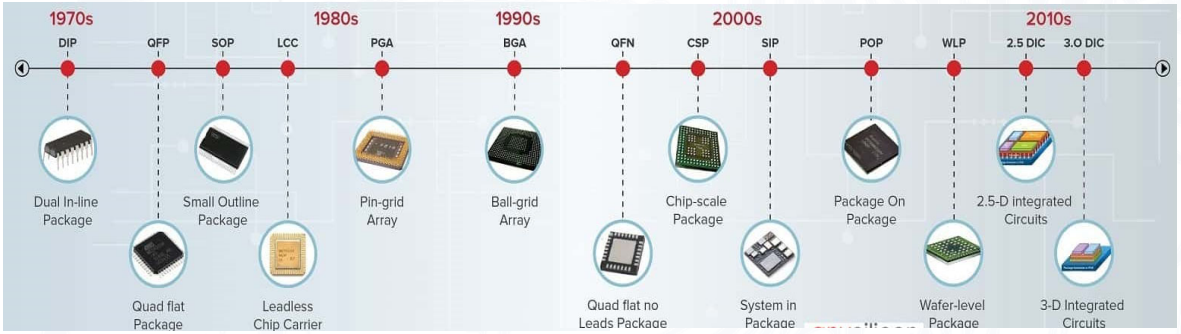
1989. Intel 80486, 8 kB cache and a floating-point math coprocessor (1 µm - 0,6 µm)

Deep–UV excimer laser lithography was commercially deployed during the 1990s

The silicon engine. A timeline of semiconductors in computers
www.computerhistory.org/siliconengine/



On 23 December **1947** two employees working at Bell Labs in the US state of New Jersey, John Bardeen and Walter Brattain, assisted by William Shockley, created the first working 'point contact transistor'.



Semiconductor packaging history trends
anysilicon.com/semiconductor-packaging-history-trends/ -

Dos patas del Tc ….hasta 2002!!!

Moore’s Law

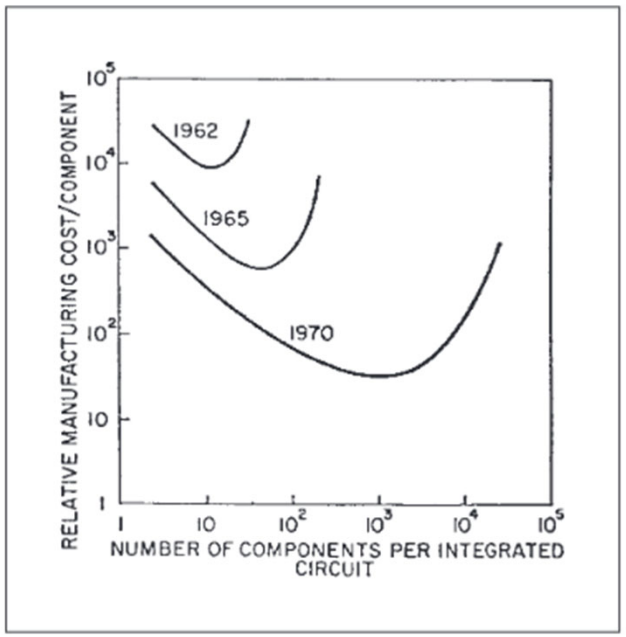
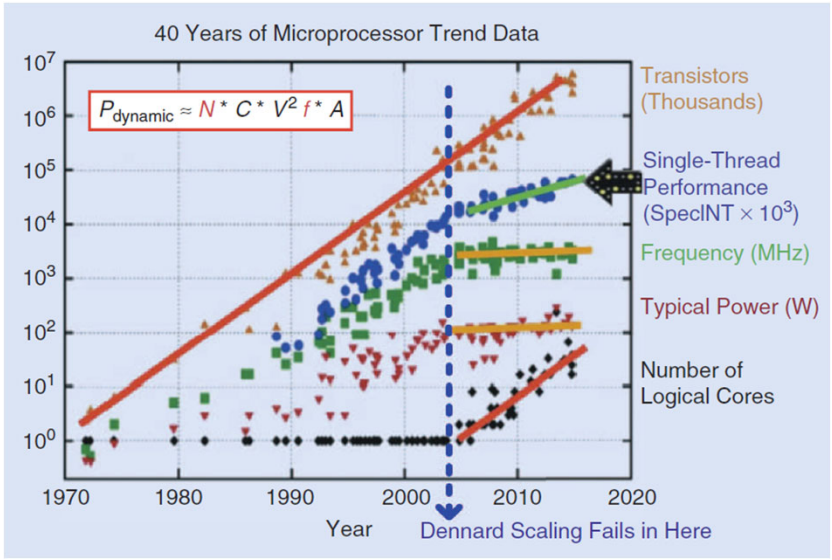


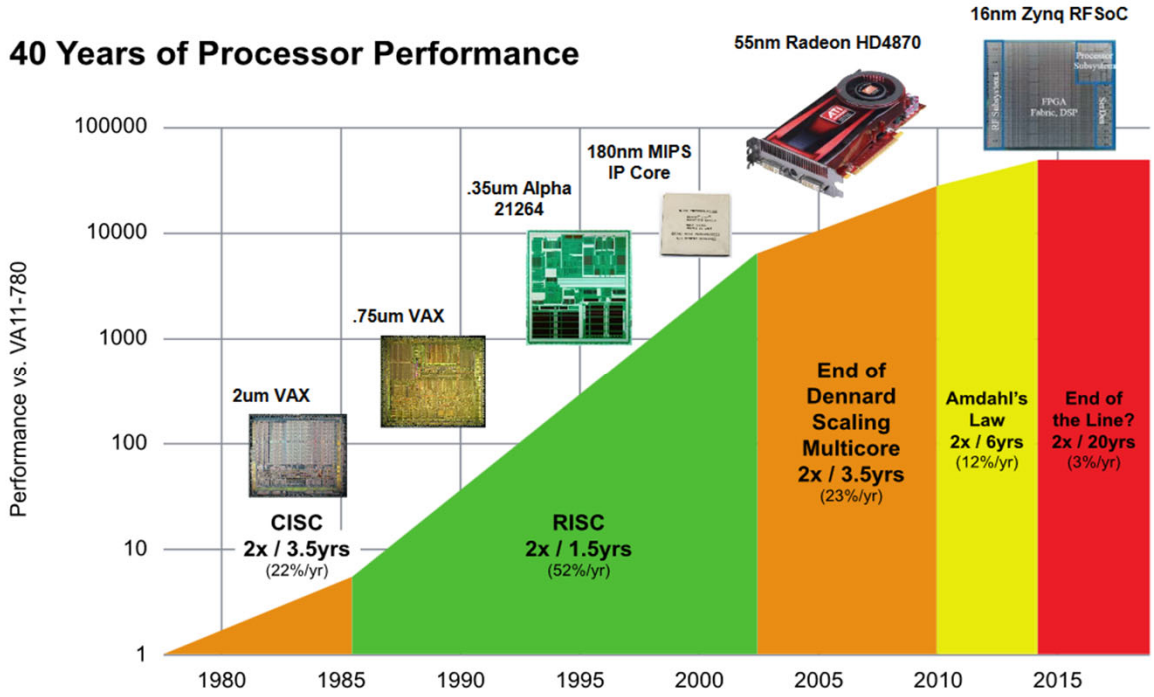
Figure 1. Gordon Moore’s original 1965 diagram illustrating the phenomenon now known as Moore’s law.

Dennard’s scaling



L. Xiu, "Time Moore: Exploiting Moore's Law From The Perspective of Time," in IEEE Solid-State Circuits Magazine, vol. 11, no. 1, pp. 39-55

Llegando al final?

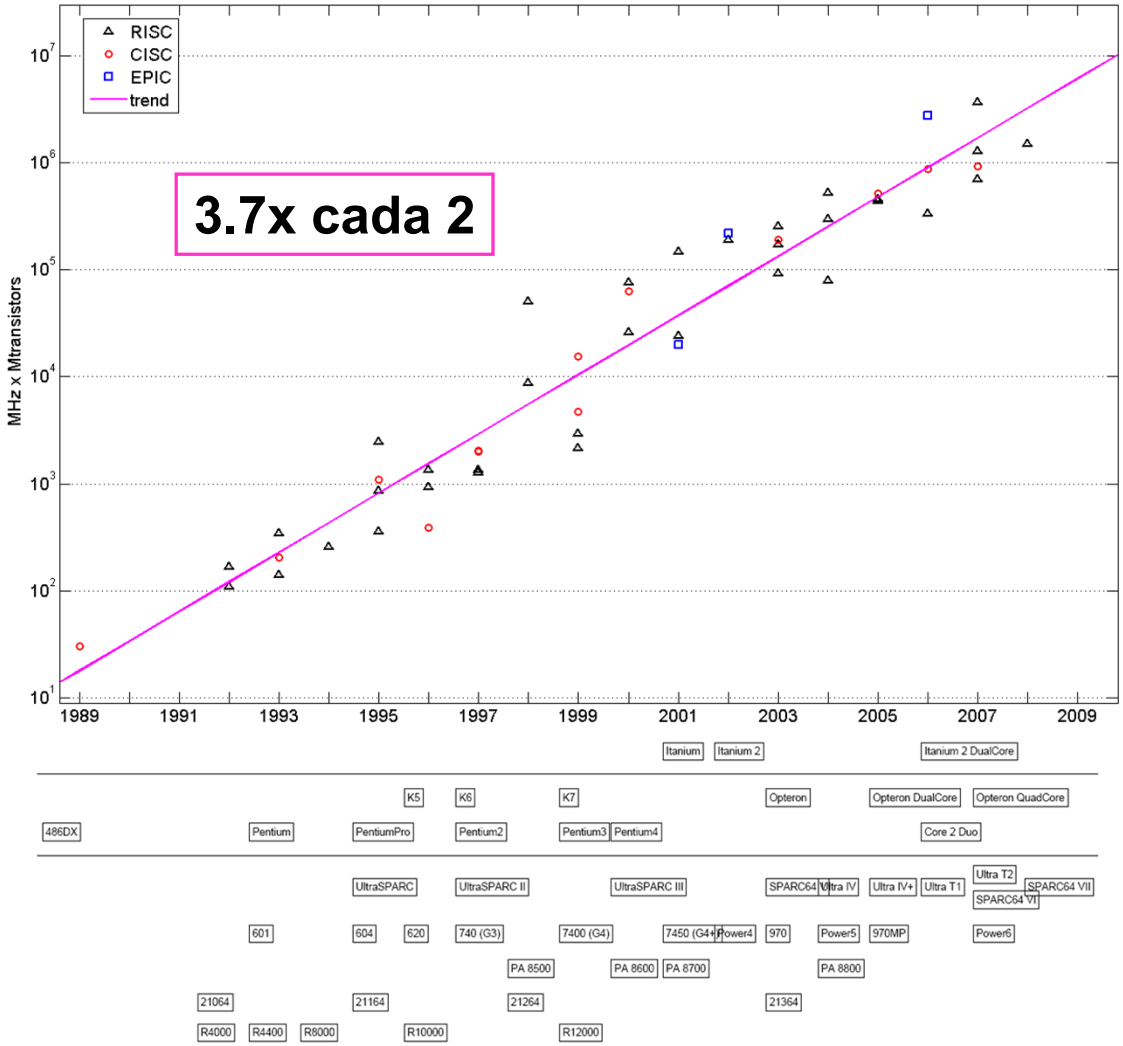


Source: John Hennessy and David Patterson, Computer Architecture: A Quantitative Approach, 6/e 2018

Moore

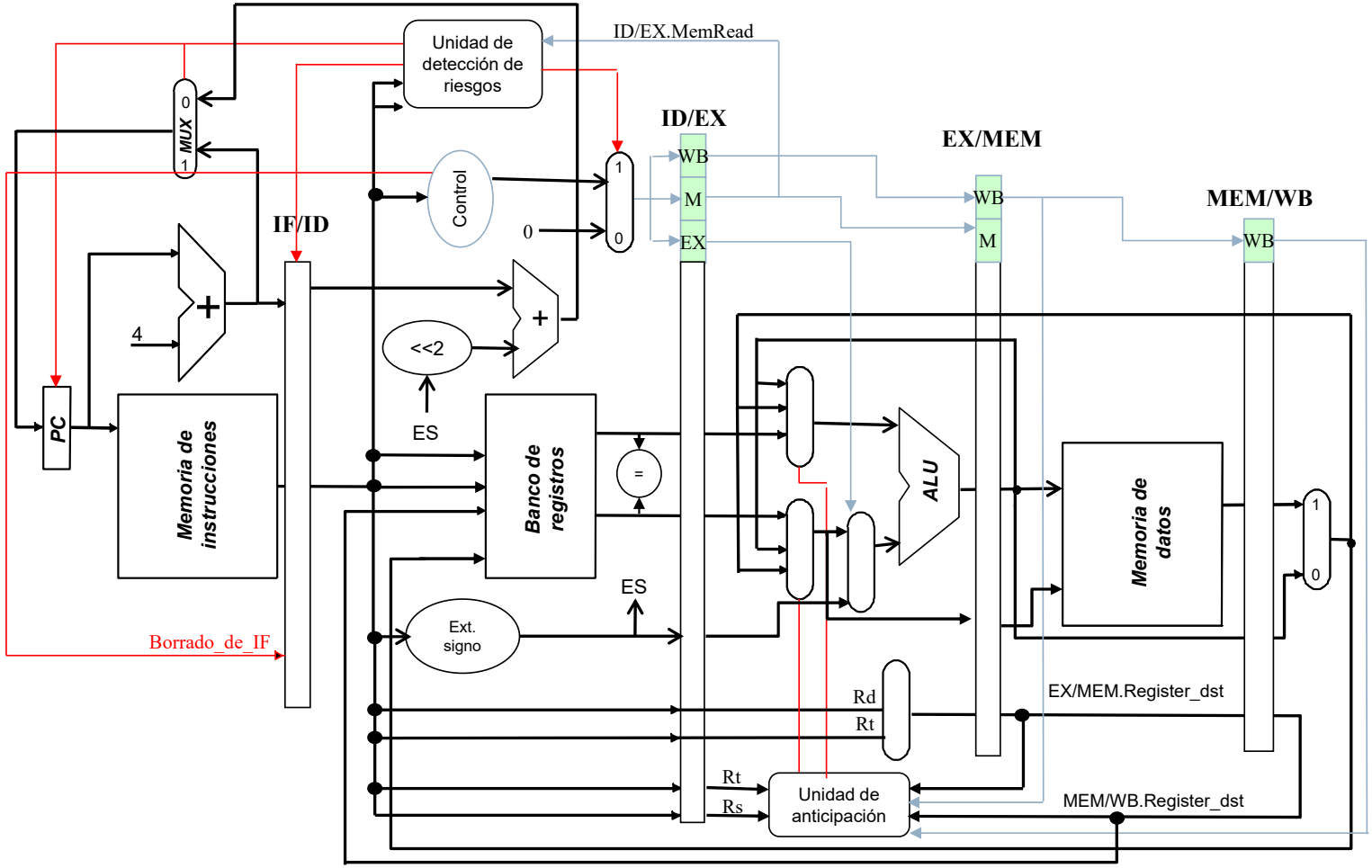
Transfreq

Actualización de:
J. Alastruey, J.L. Briz, P. Ibáñez y V. Viñals. “Software Demand vs. Hardware Supply: SPEC CPU and processor resources”. *IEEE Micro*, IEEE Computer Society 2006.



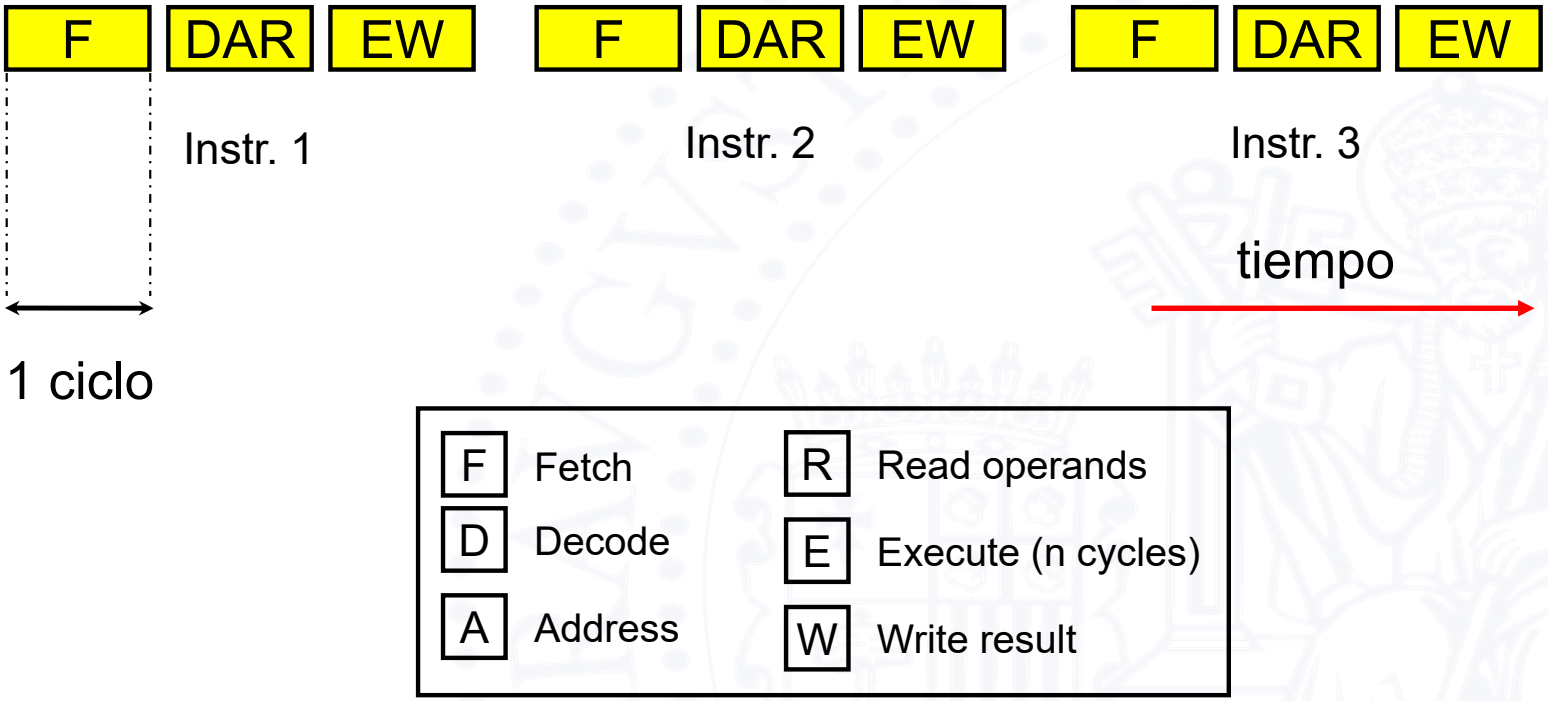
La tercera pata del Tc....
CPI – IPC^{-1}

MIPS base con detención y anticipación



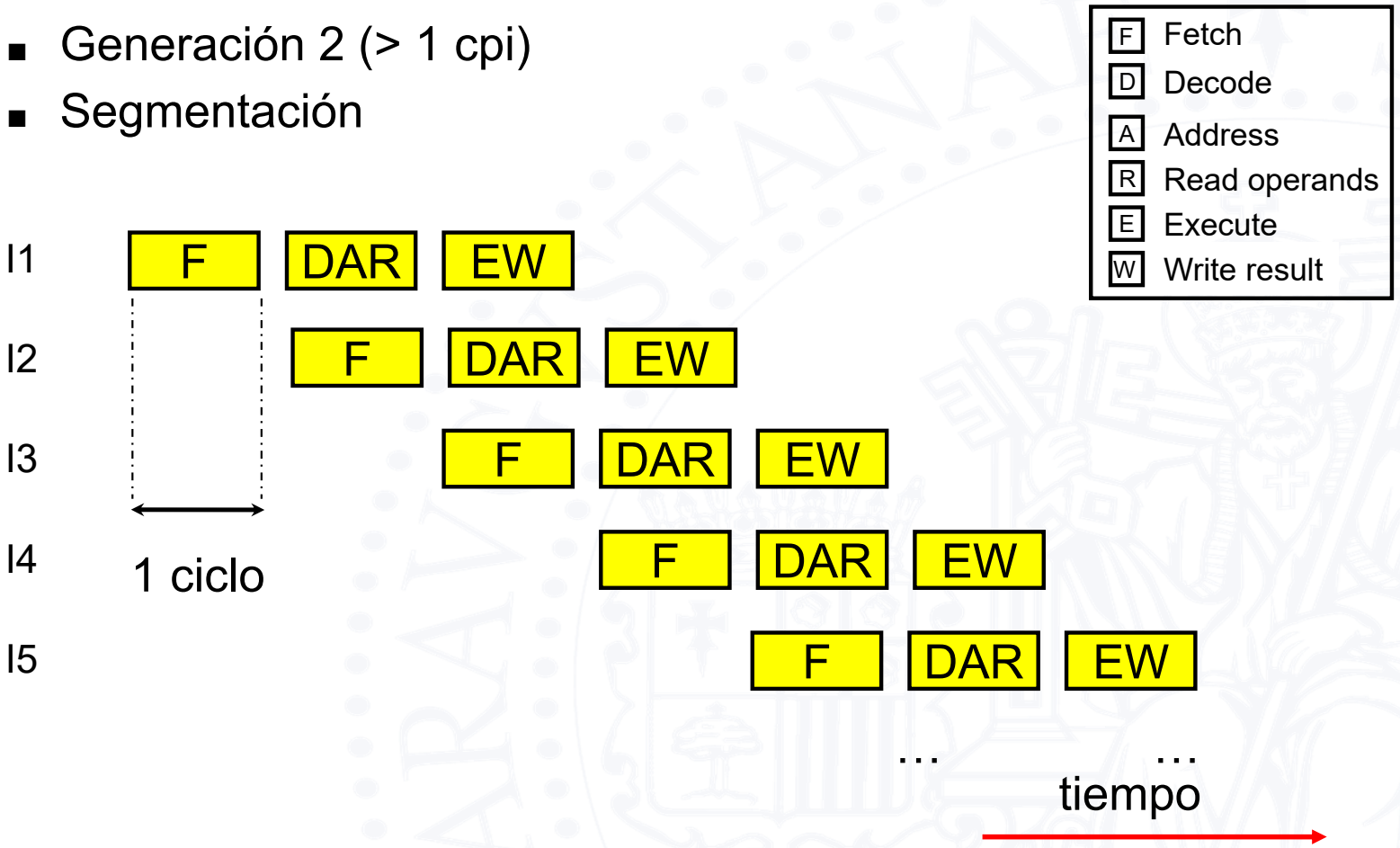
Modelo de ejecución y rendimiento

- Generación 1 (> 3 **c**iclos **p**or **i**nstrucción)



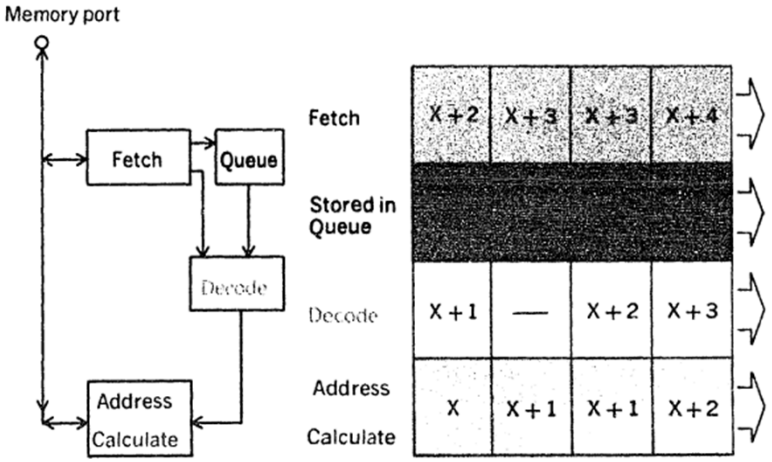
Evolución en el modelo de ejecución

- Generación 2 (> 1 cpi)
- Segmentación



8086

Pipeline sí… pero poco



Solid state/computers

New options from big chips

Large microprocessor dice mean circuit designers must design 'smarter'

J. McKevitt and J. Bayliss, "New options from big chips," in IEEE Spectrum, vol. 16, no. 3, pp. 28-34, March 1979

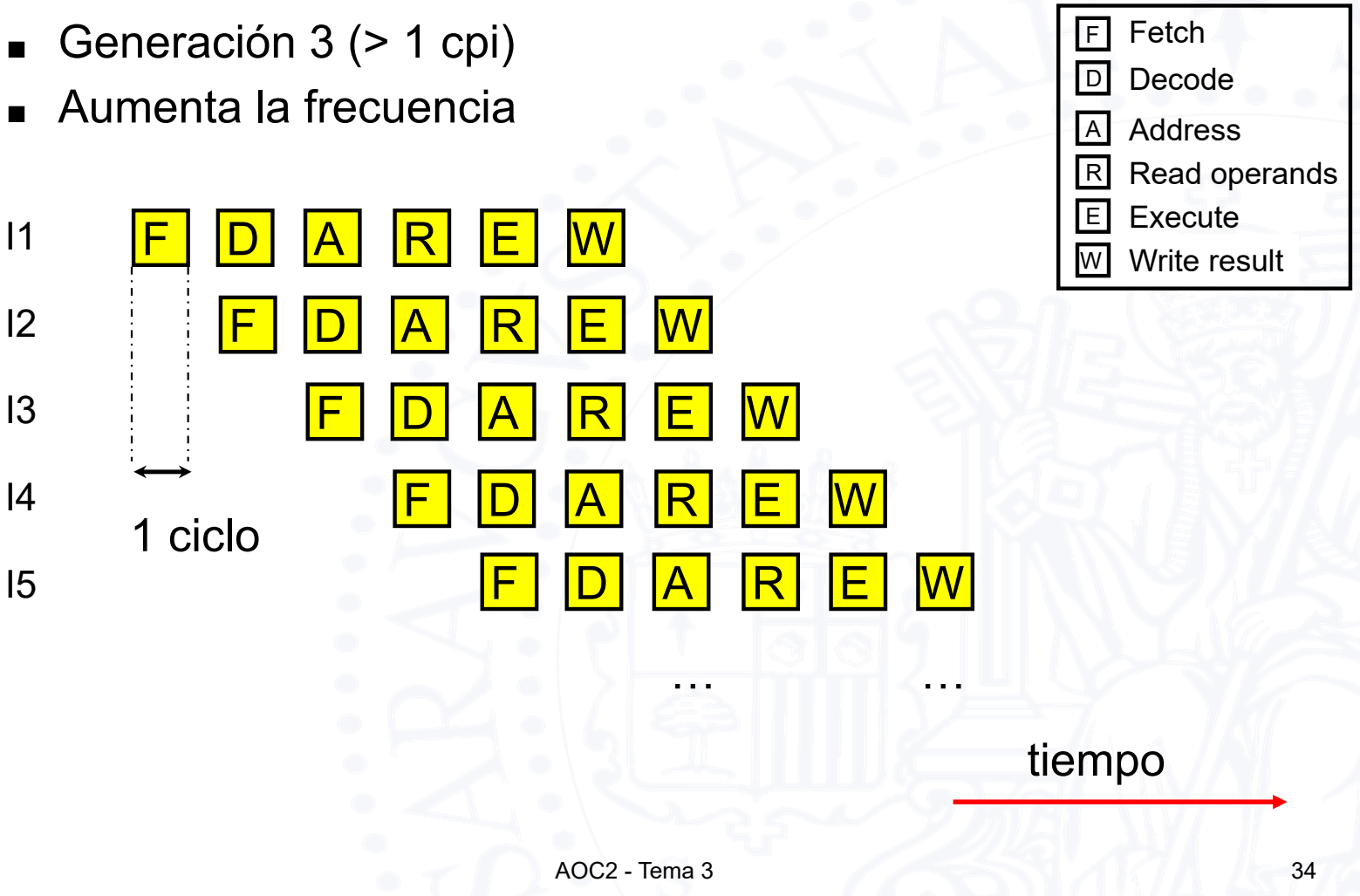
- Intentaron segmentaciones más agresivas
- Problemas:

"65 percent more die area to extract the next rise of 22 percent in performance"

"The cost is even higher for MOS technology. Hardware costs do not rise linearly with hardware area. They are subject to MOS production yield curves"

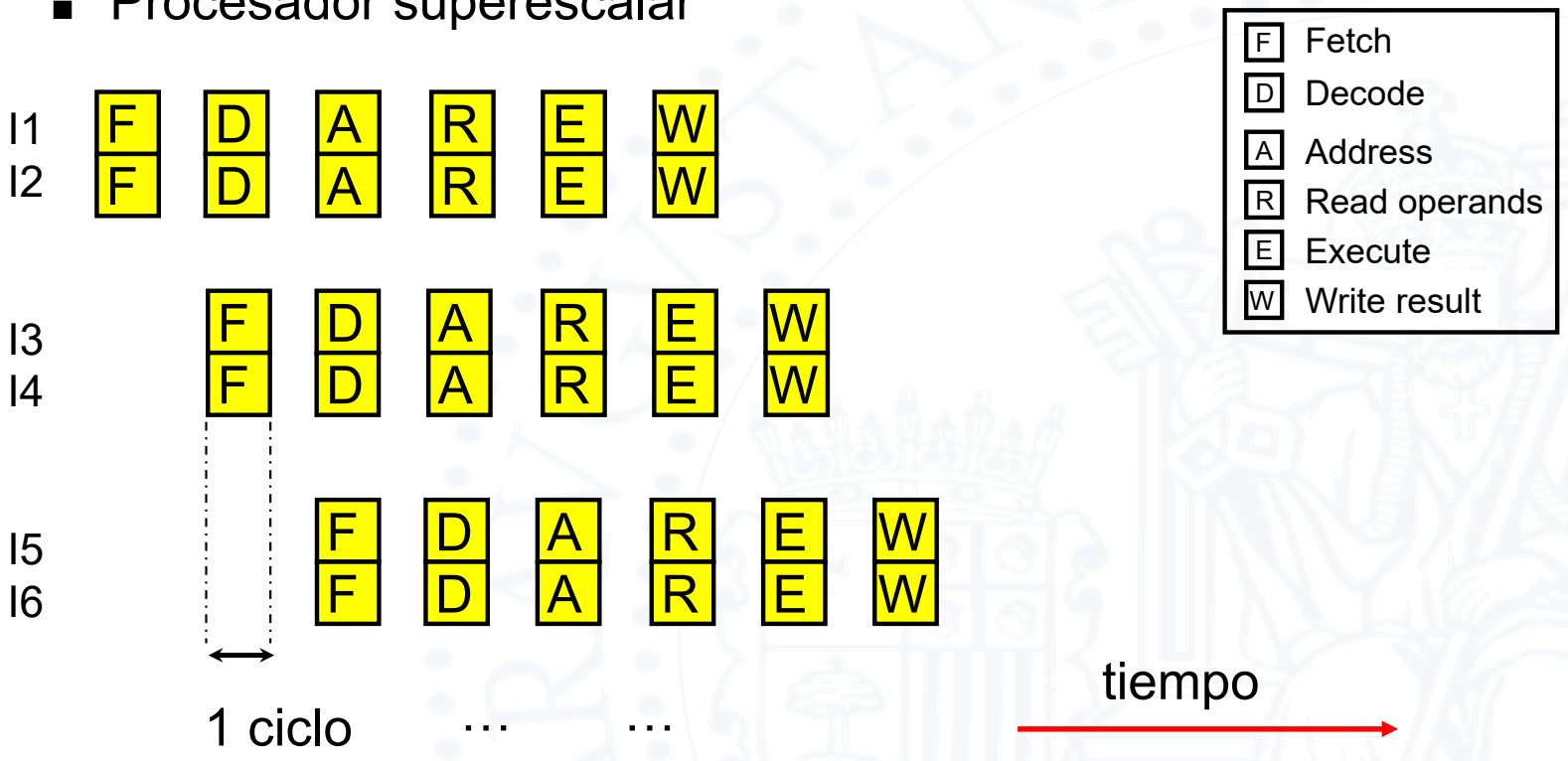
Evolución en el modelo de ejecución

- Generación 3 (> 1 cpi)
- Aumenta la frecuencia



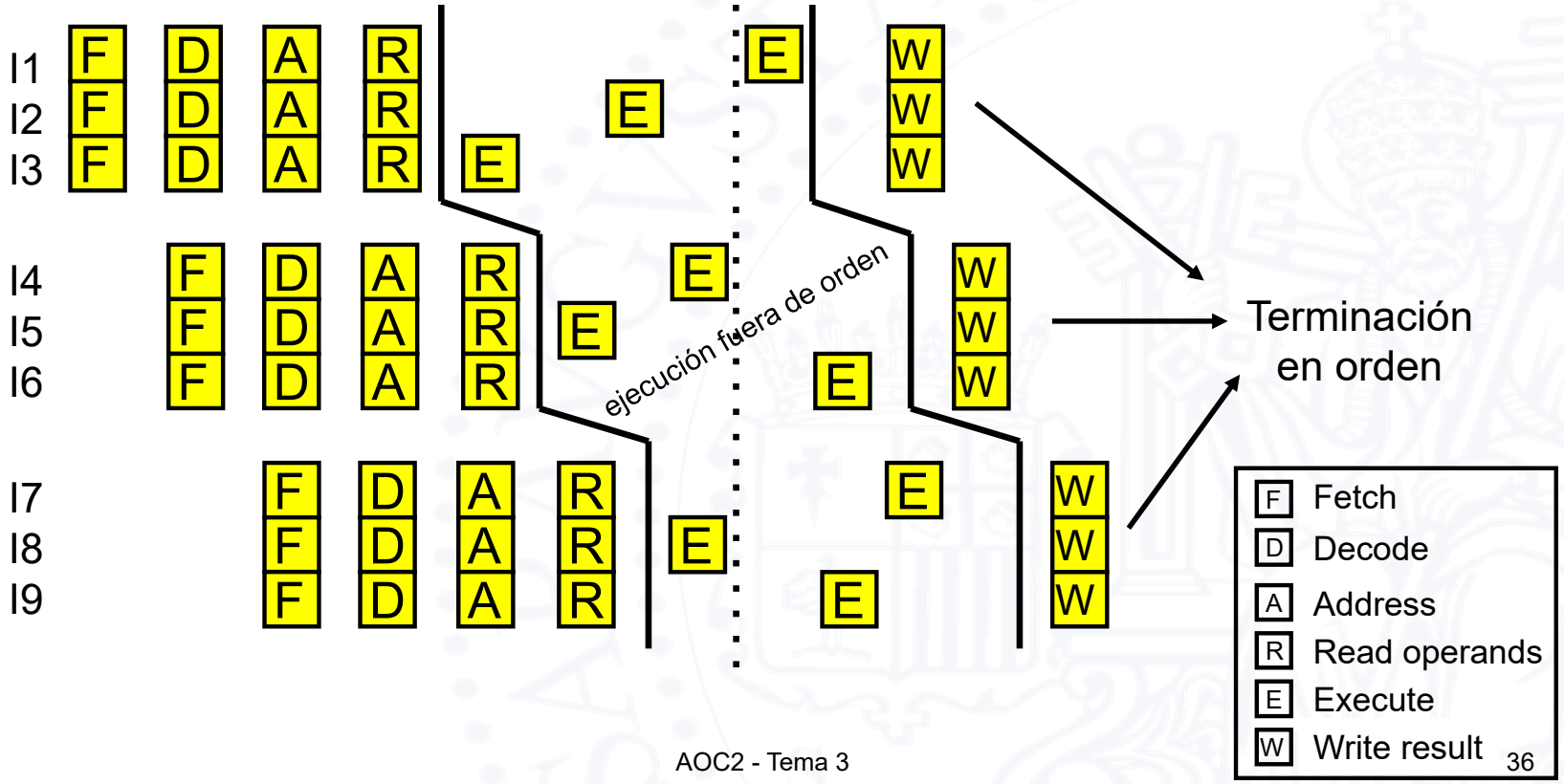
Evolución en el modelo de ejecución

- Generación 4 (> 0.5 cpi) – en μ a finales de los 80 –
- Procesador superescalar



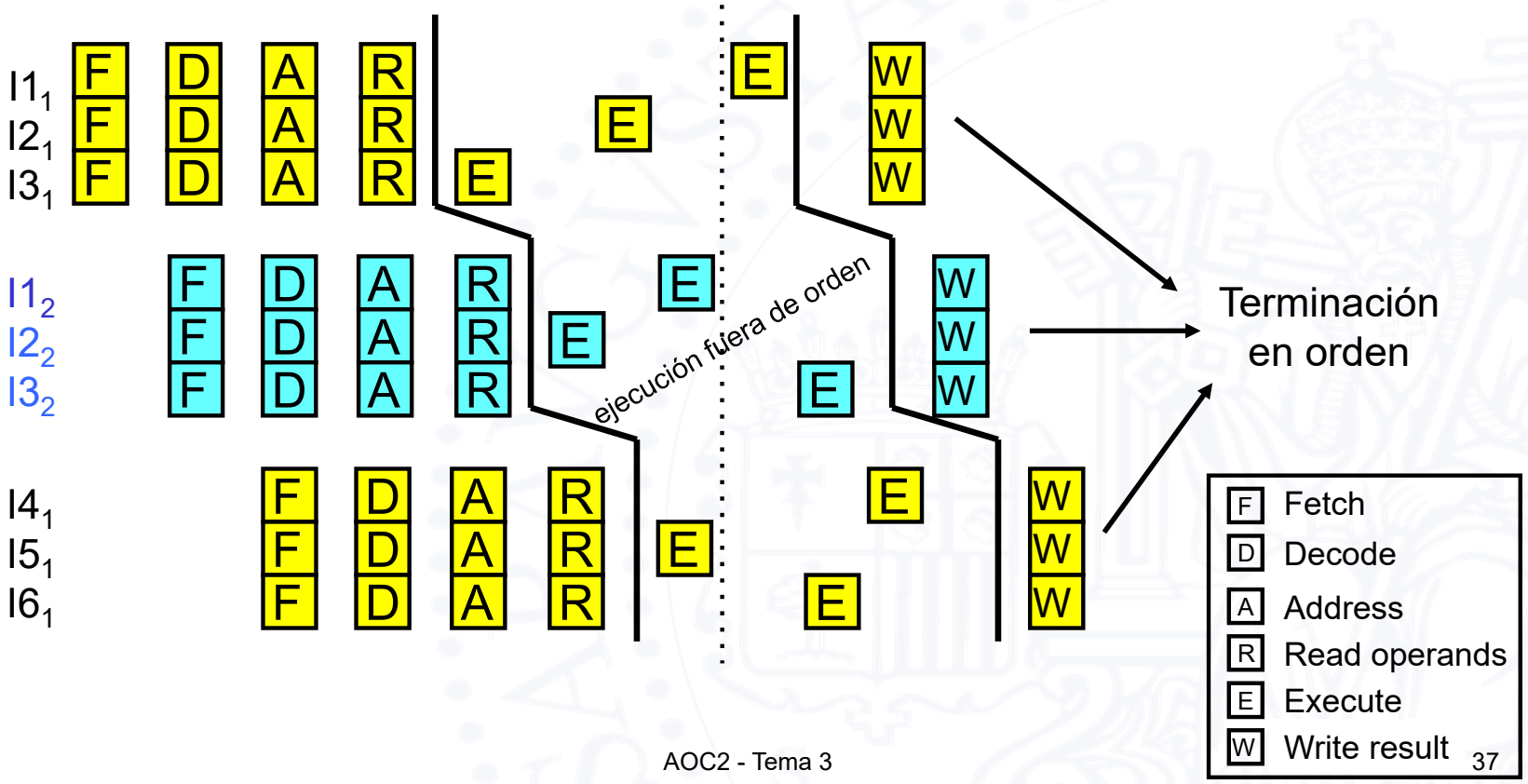
Evolución en el modelo de ejecución

- Generación 5 (≈ 0.3 cpi) – en μ a principios de los 90 –
- Ejecución en desorden



Evolución en el modelo de ejecución

- Generación 6 (≈ 0.25 cpi) – en μ en los años 2000 –
- Multithreading



Paralelismo por replicación

Tipos de interconexión

Interconnection network	Dimensions	Maximum system bandwidth	Bisection bandwidth	Wire count per data bit
Bus	0	C	C	1
Ring	1	P	C	P
2D mesh	2	P	$\sqrt{P}$	$4P$
3D mesh	3	P	$\sqrt[2/3]{P}$	$6P$

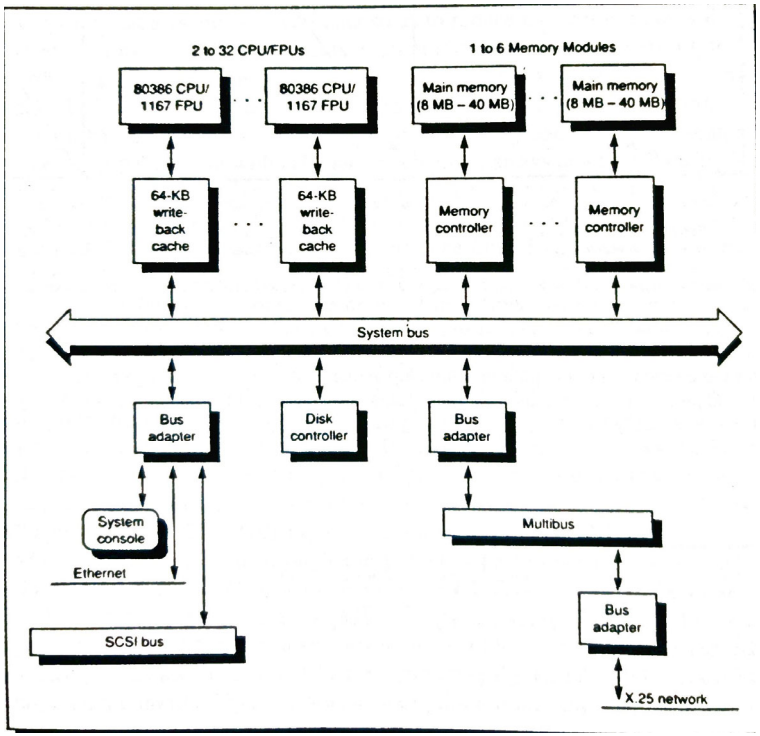
Tipos de comunicación

- DSM
- SMP
- Paso mensajes

Método de comunicación

DSM
SMP
Paso mensajes

SMP



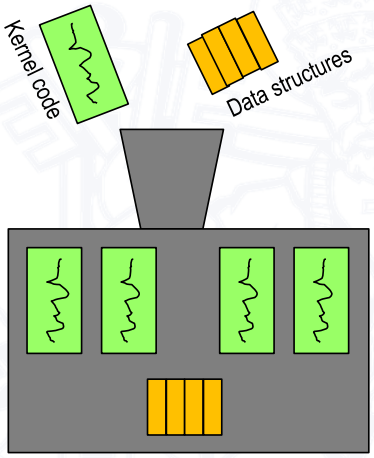
Sequent Symmetry multiprocessor

- Hasta 30 micros sobre bus de sistema
- Cache privada off-chip
- Jerarquía de buses para IO

GPGPUs – TLP masivo

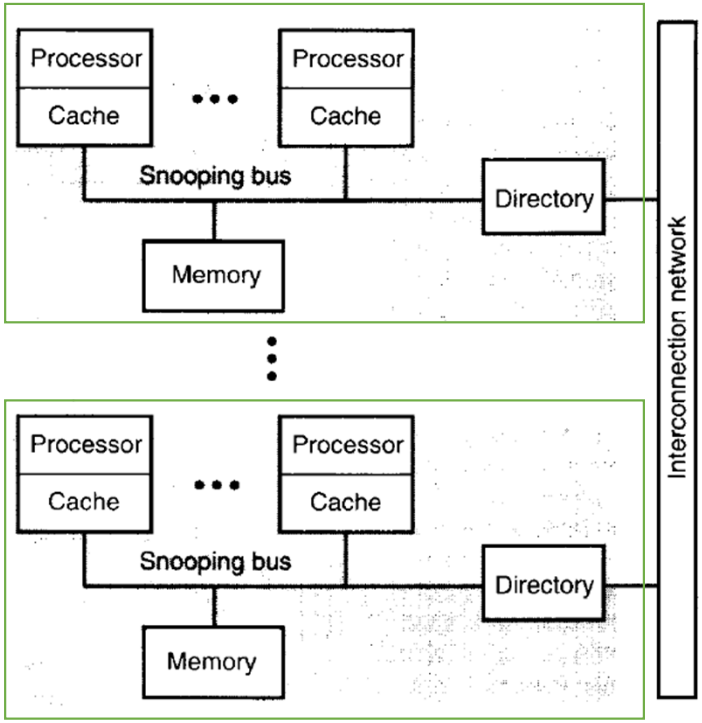
- SIMD / SIMP (CUDA/OpenCL)
  - Ejemplo: C = A + B

```
__global__ void matAdd(float A[N][N], float B[N][N],float C[N][N])
{
        int i = threadIdx.x;
        int j = threadIdx.y;
        C[i][j] = A[i][j] + B[i][j];
}

int main()
{
// Allocate structures in device
// Transfer data to device
        …
// Set grid parameters and number of threads
        dim3 dimBlock(N, N);
// Kernel invocation
        matAdd<<<1, dimBlock>>>(A, B, C);
// Transfer results from device
…
}
```



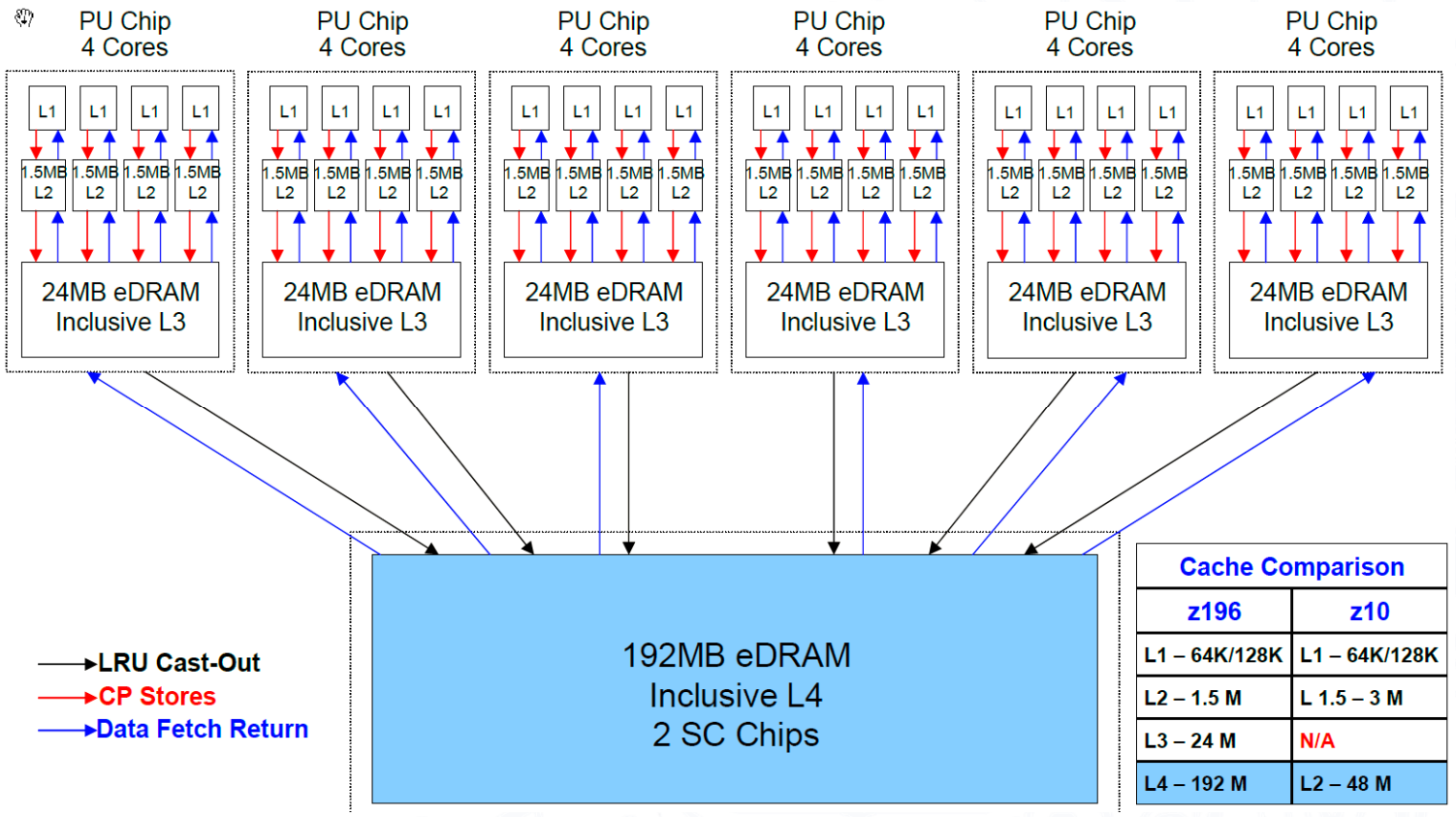
Nvidia device

DSM



D. Lenoski et al., "The Stanford Dash multiprocessor," in Computer, vol. 25, no. 3, pp. 63-79, March 1992

z196 book-level cache hierarchy (24 cores)



Cache Comparison	
z196	z10
L1 – 64K/128K	L1 – 64K/128K
L2 – 1.5 M	L 1.5 – 3 M
L3 – 24 M	N/A
L4 – 192 M	L2 – 48 M

z10 in-order processor pipeline

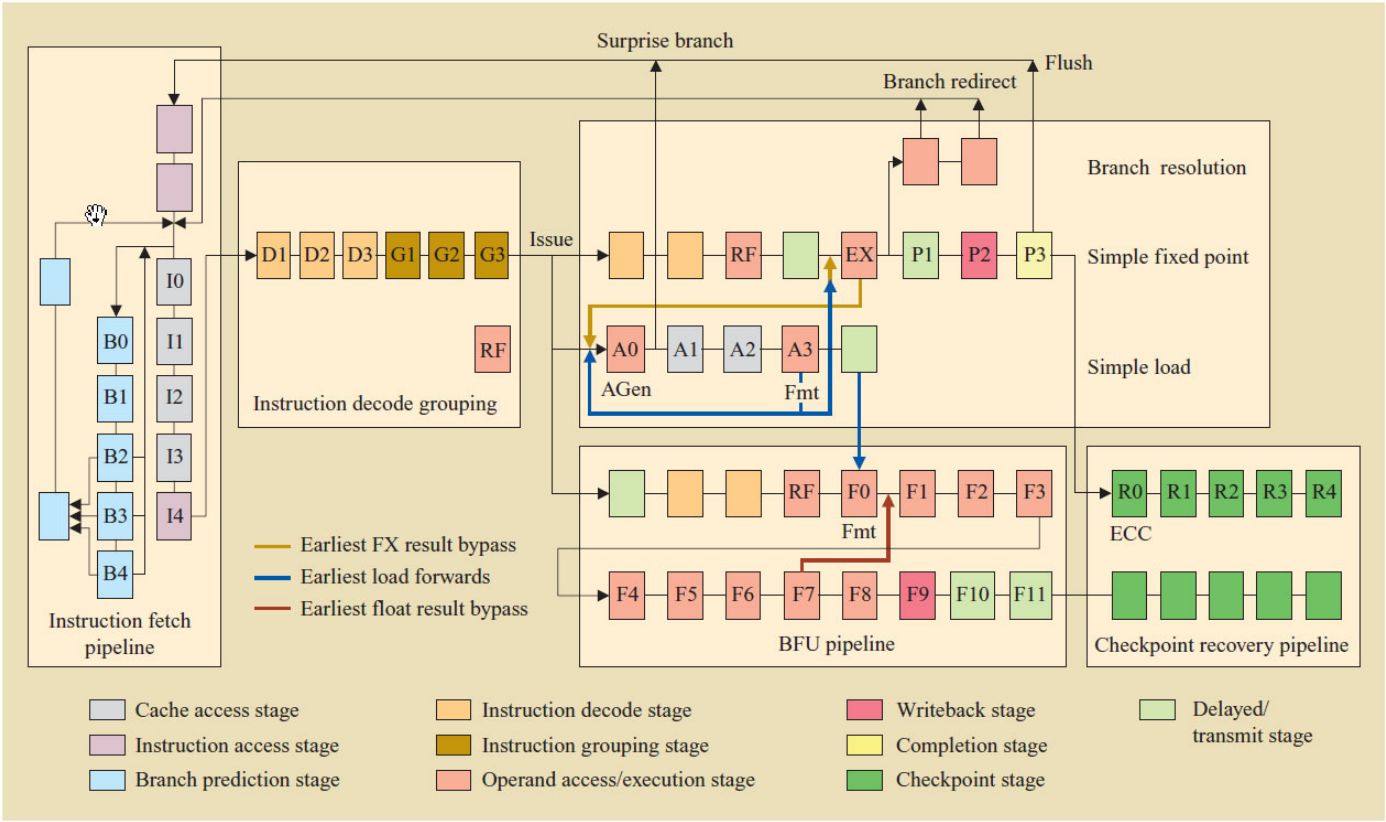
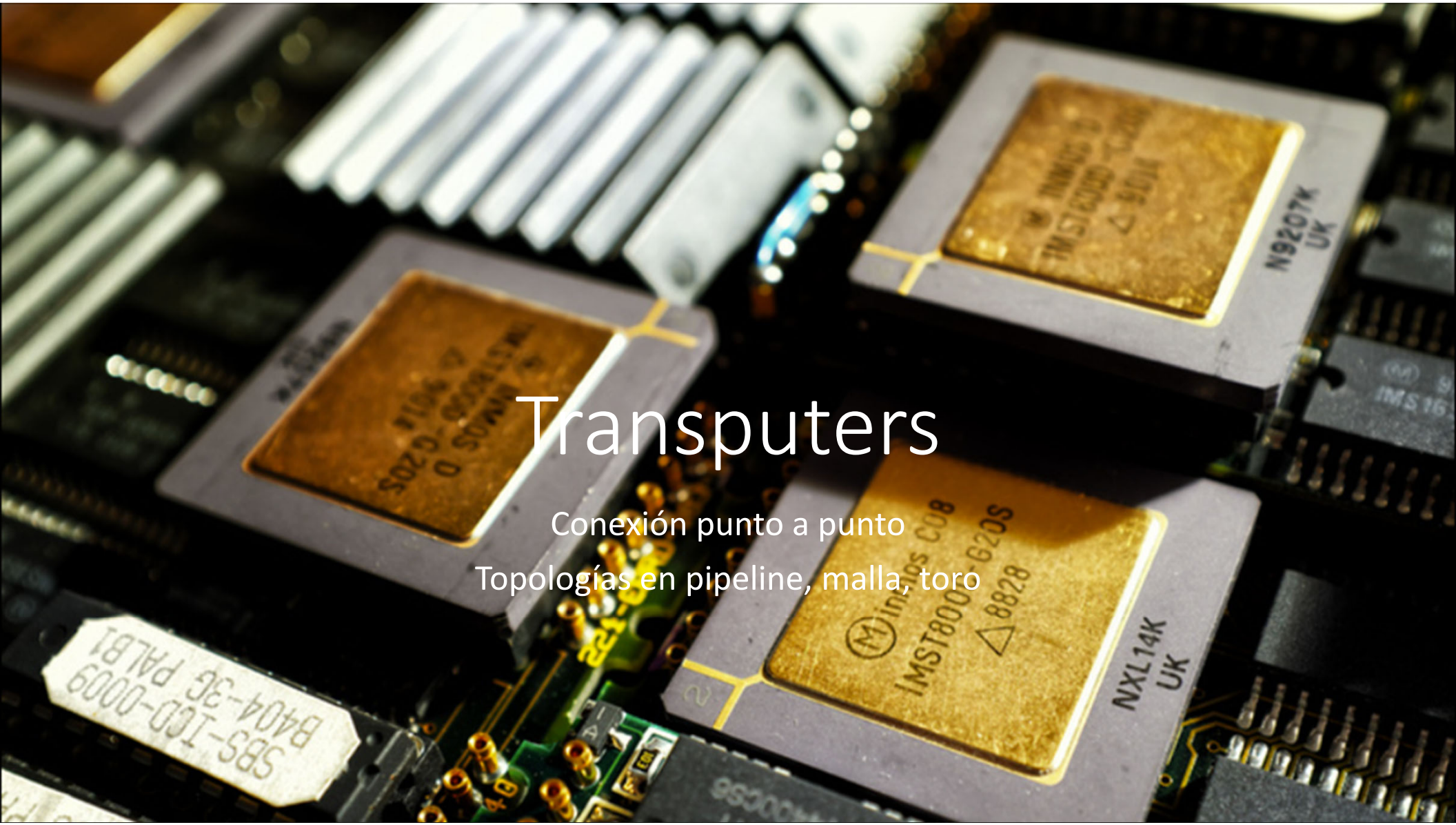


Figure 3

The z10 microprocessor pipeline. (AGen: address generation; BFU: binary floating-point unit; ECC: error-correcting code; EX: execution; Fmt: format; RF: general-purpose register [GPR] file/floating-point register [FPR] file access.)

C.-L. K. Shum, F. Busaba, S. Dao-Trong, G. Gerwig, C. Jacobi, T. Koehler, E. Pfeffer, B. R. Prasky, J. G. Rell, and A. Tsai. 2009. Design and microarchitecture of the IBM system z10 microprocessor. *IBM J. Res. Dev.* 53, 1 (January 2009), 1-12.


Transputers
Conexión punto a punto
Topologías en pipeline, malla, toro
N9207K
UK
IMST800-G20S
8828
NXL14K
UK

Paralelismo por replicación

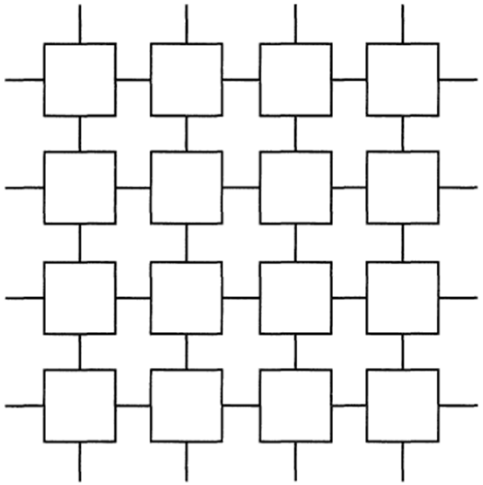
Tipos de interconexión

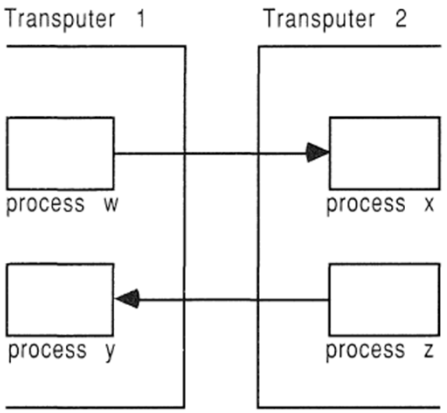
Interconnection network	Dimensions	Maximum system bandwidth	Bisection bandwidth	Wire count per data bit
Bus	0	C	C	1
Ring	1	P	C	P
2D mesh	2	P	$\sqrt{P}$	$4P$
3D mesh	3	P	$\sqrt[2/3]{P}$	$6P$

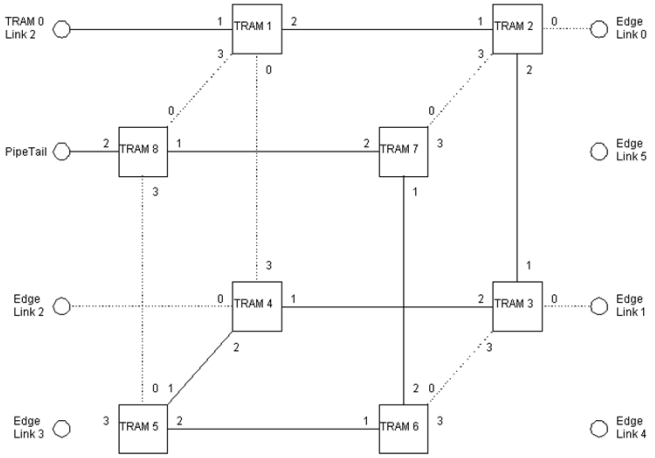
Tipos de comunicación

- DSM
- SMP
- Paso mensajes

Topologías



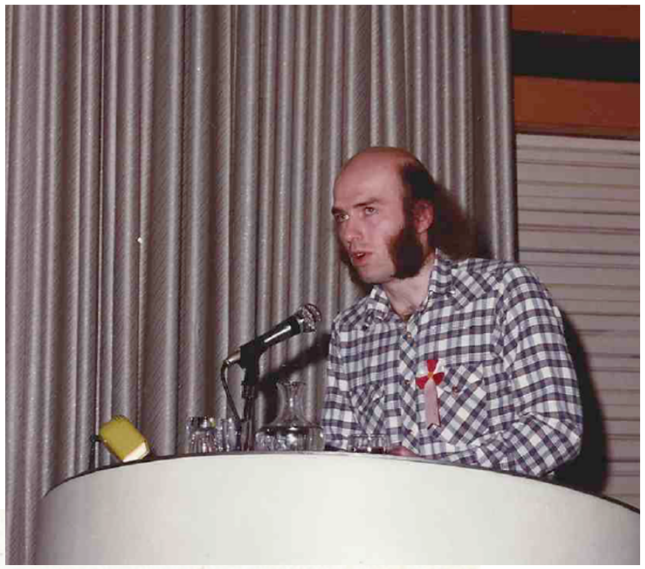

Transputer 1
Transputer 2
process w
process x
process y
process z

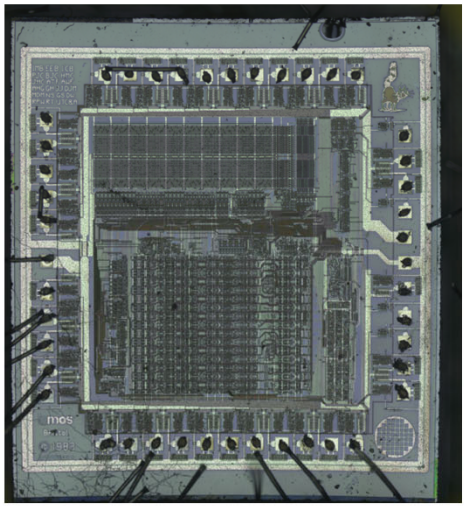

TRAM 0
Link 2
TRAM 1
TRAM 2
Edge
Link 0
PipeTail
TRAM 8
TRAM 7
Edge
Link 5
Edge
Link 2
TRAM 4
TRAM 3
Edge
Link 1
Edge
Link 3
TRAM 5
TRAM 6
Edge
Link 4

Simple 42 (1982)

1. Introduction

This document describes the Simple 42. This is the machine currently being implemented.





1. Simple 42 Documentation

The following documents are the current state of our efforts to document the Simple 42 implementation. They are not complete. They describe:

The microinstruction ROM
The data path
The minor cycle timing
The memory interface control
The microcode assembly language
The microcode

Guy Harriman
David May

17.3.82

SUNDAY TIMES Sunday 23 July 1978

Debut of the mighty micro men

By James Poole

Meet the micro men Iann Barron, Dr. Dick Petritz, and Dr. Paul Schroeder, founders of Inmos, at the door of the NEB.

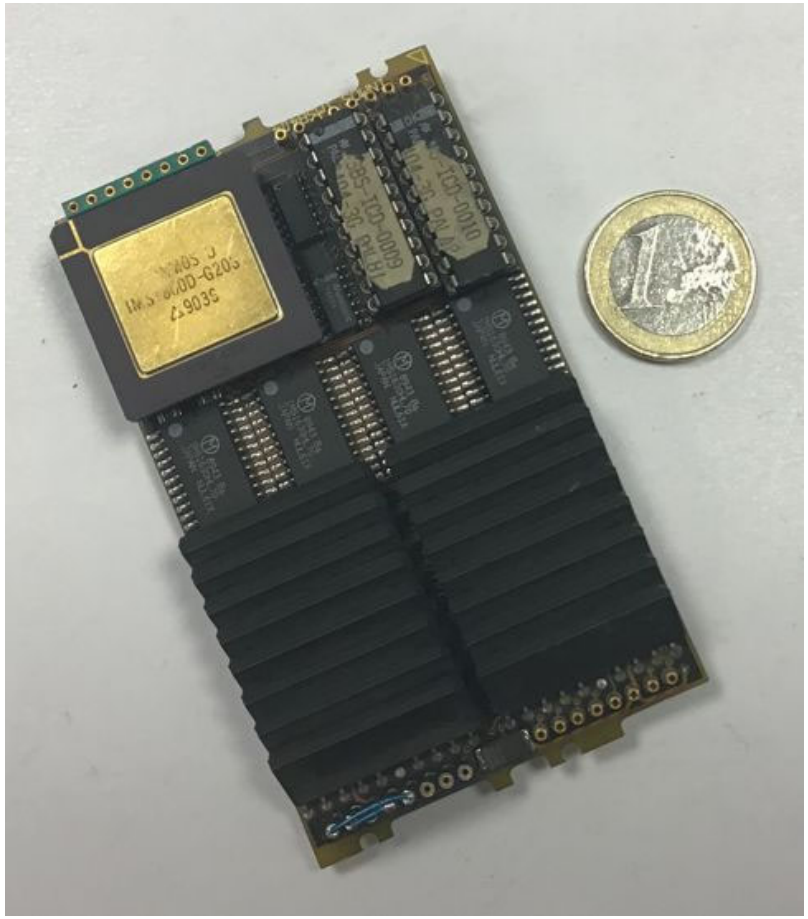
Historia sucinta: CSP y robot controllers…

- 1978: Communicating Sequential Processes (Hoare); EPL (May)
- 1983: First occam 1 programming manual
- 1984: Drafts of occam 2 language defintion
- February 1984: ACM Sigplan occam article
- September 1984: Demo of Simple 42, Occam User Group, Bristol
- November 1984: occam programming system (VAX) and portakit
- Hoare and Roscoe, Programs are Predicates, Tokyo, 1984

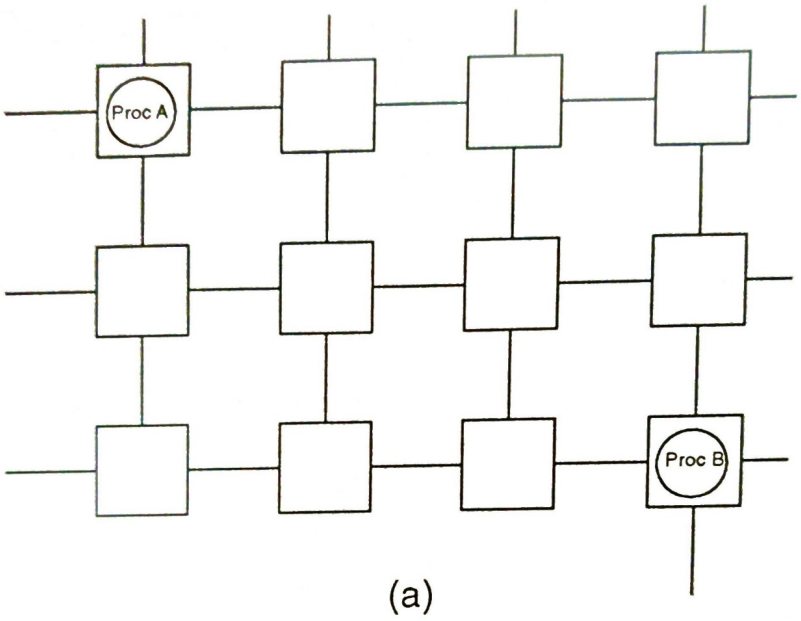
Historia sucinta: Inmos & Transputer

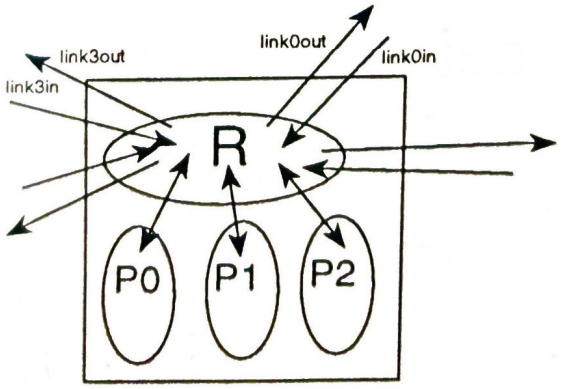
- 1978: Inmos founded with £50 million backing from UK government
- 1979: Operations start in Colorado Springs and Bristol
- 1979: Bristol team grows to about 50; average age about 25
- 1979-83: occam, transputer architecture, CAD system, prototype
- 1982-83: first articles about occam and transputer published
- 1982: Inmos Newport factory (Richard Rogers partnership)
- Japanese MITI starts $1billion fifth generation project; founds ICOT

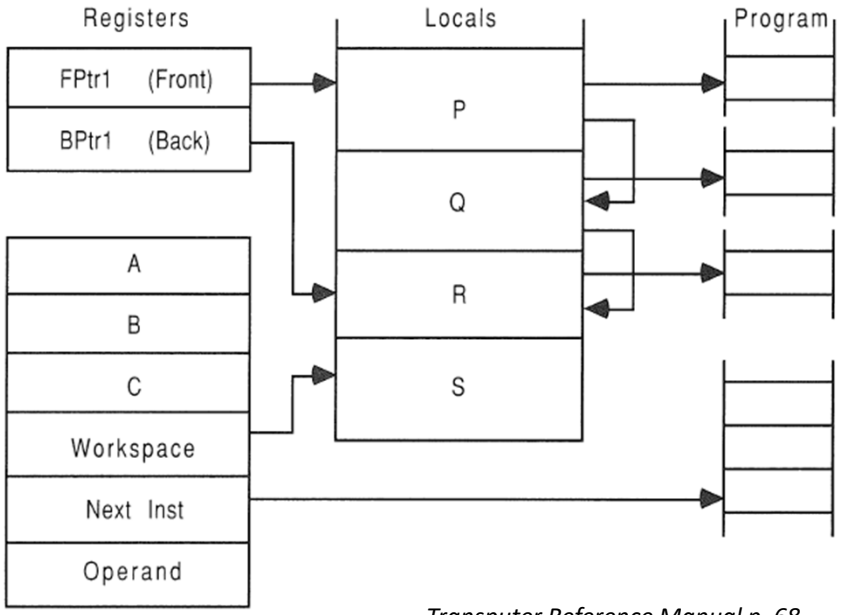
T800



- Registros 32b
- 5 / 10 / 20 [MHz | MIPS]
- 2.5 Mflops Livermoore loop 7
- 4KB SRAM on-chip
- DRAM en placa hasta 4 GB
- IO mapeada + 4 links
- K_{DMA}



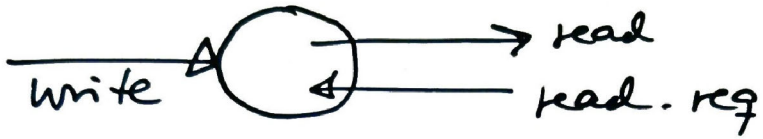




Transputer Reference Manual p. 68

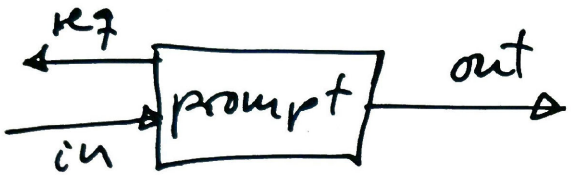
- 1 Proceso de alta prioridad (hasta fin o bloqueo)
- Resto baja prioridad, RR (quantum)
- Context switch and IRQ lat < 1 us (T800)

SINGLE BUFFER (MEMORY CELL)



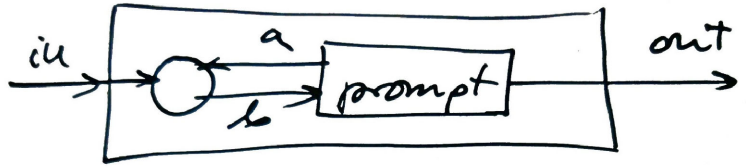

```
PROC mem.cell (CHAN OF INT  write,
               CHAN OF BOOL read.req,
               CHAN OF INT  read)
  -- Problems?
INT x:
WHILE TRUE
  ALT
    write? x
      SKIP
    BOOL any:
    read.req ? any
      read ! x
:
```

PROMPT

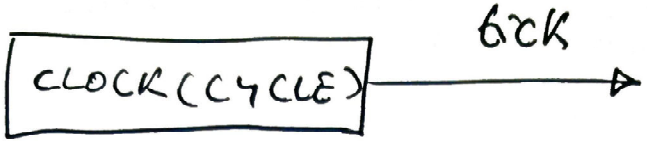



```
PROC prompt ( CHAN OF INT in,
              CHAN OF BOOL req,
              CHAN OF INT out )
  WHILE TRUE
    INT x:
    SEQ
      req ! TRUE
      in ? x
      out ! x
:
```

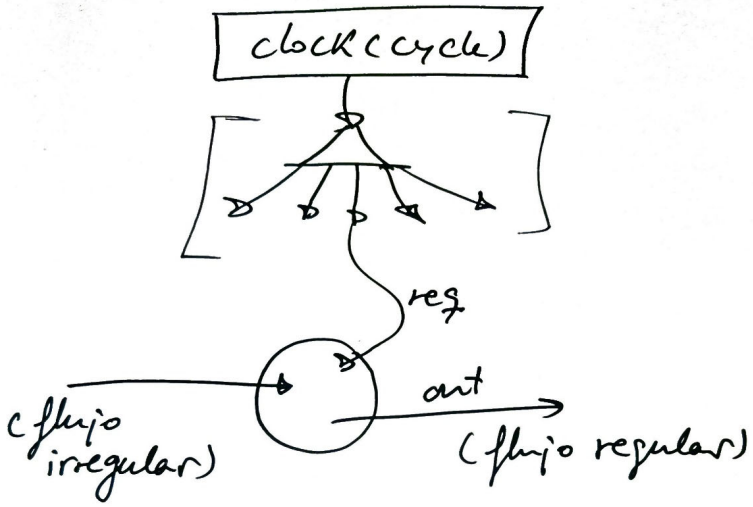
ASYNC. CELL




```
PROC async ( CHAN OF in, out)
  CHAN OF INT b:
  CHAN OF BOOL a:
  PAR
    mem. cell (in, a, b)
    prompt (b, a, out)
:
```

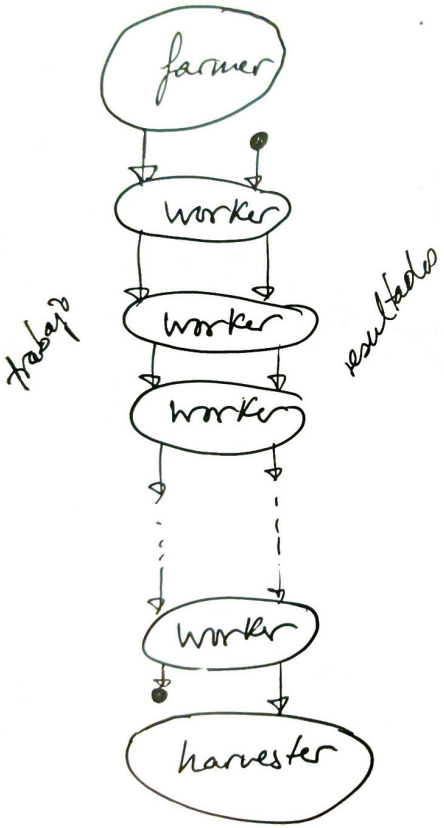



```
PROC clock (VAL INT cycle,
            CHAN OF BOOL tick)
  TIMER tim:
  INT t:
  SEQ
    tim ? t
    WHILE TRUE
      SEQ
        t := t PLUS cycle  -- u
        tim ? AFTER t
        tick ! TRUE
```

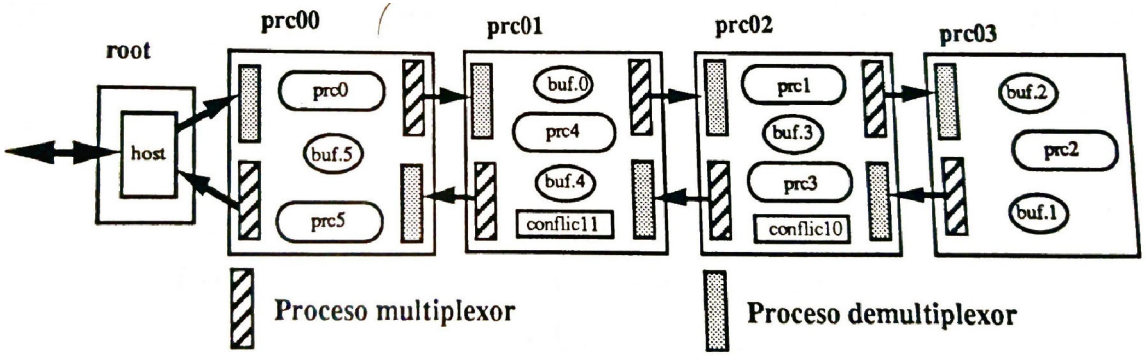



```
PRI PAR
  SEQ
    clock
    [demux]
      mem.all
  PAR
    .
    .
    .
```

PROCESS FARMING

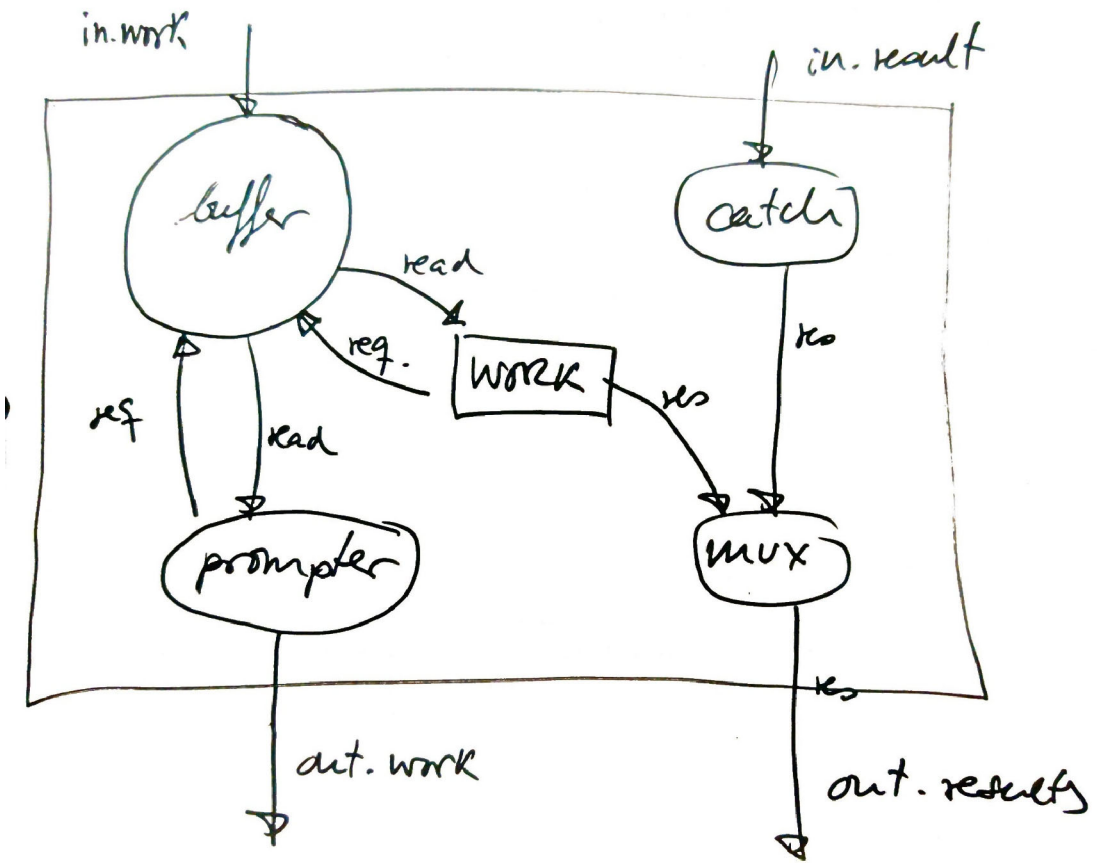


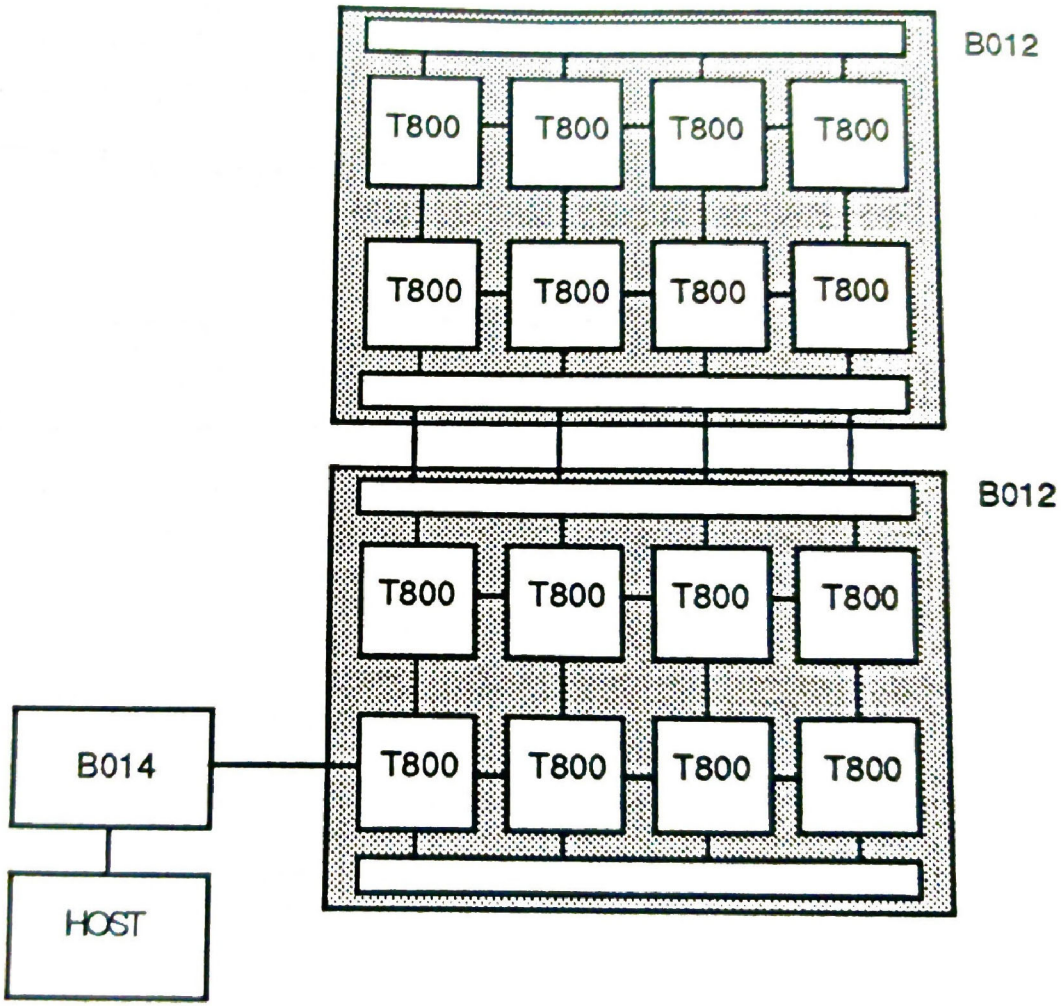
PPO 2



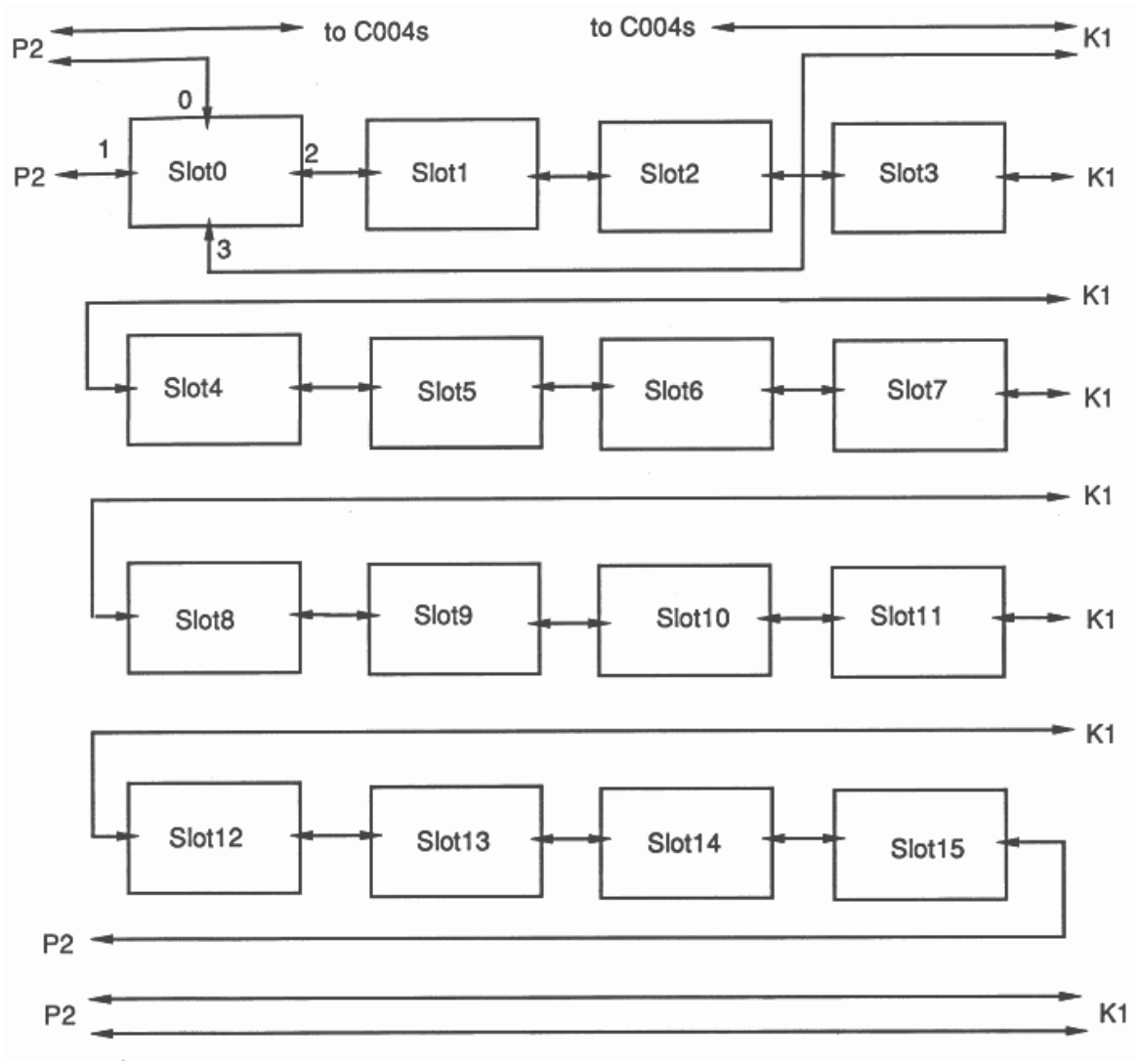
dato? x
si tengo curro
paso ! x
sino
r = proceso(x)
resul ! r

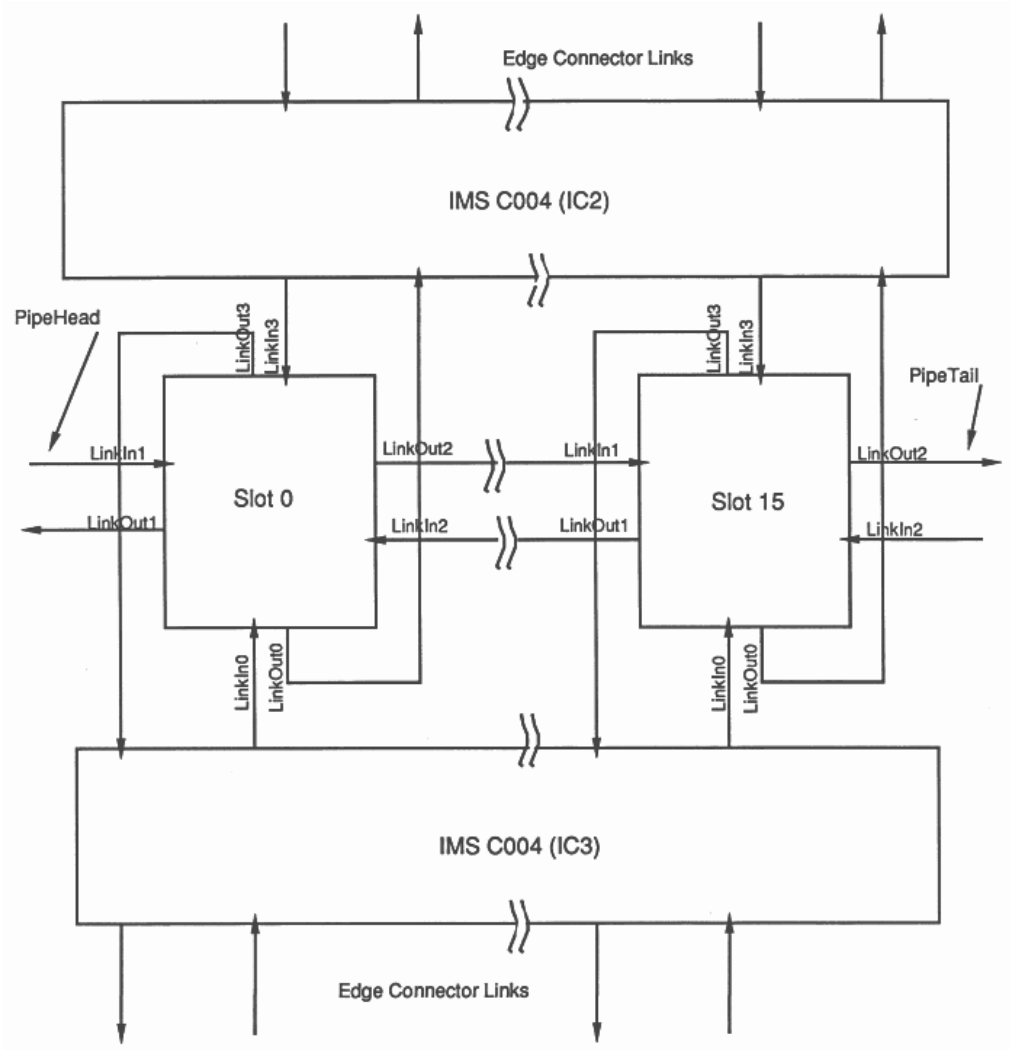
Estructura de un worker

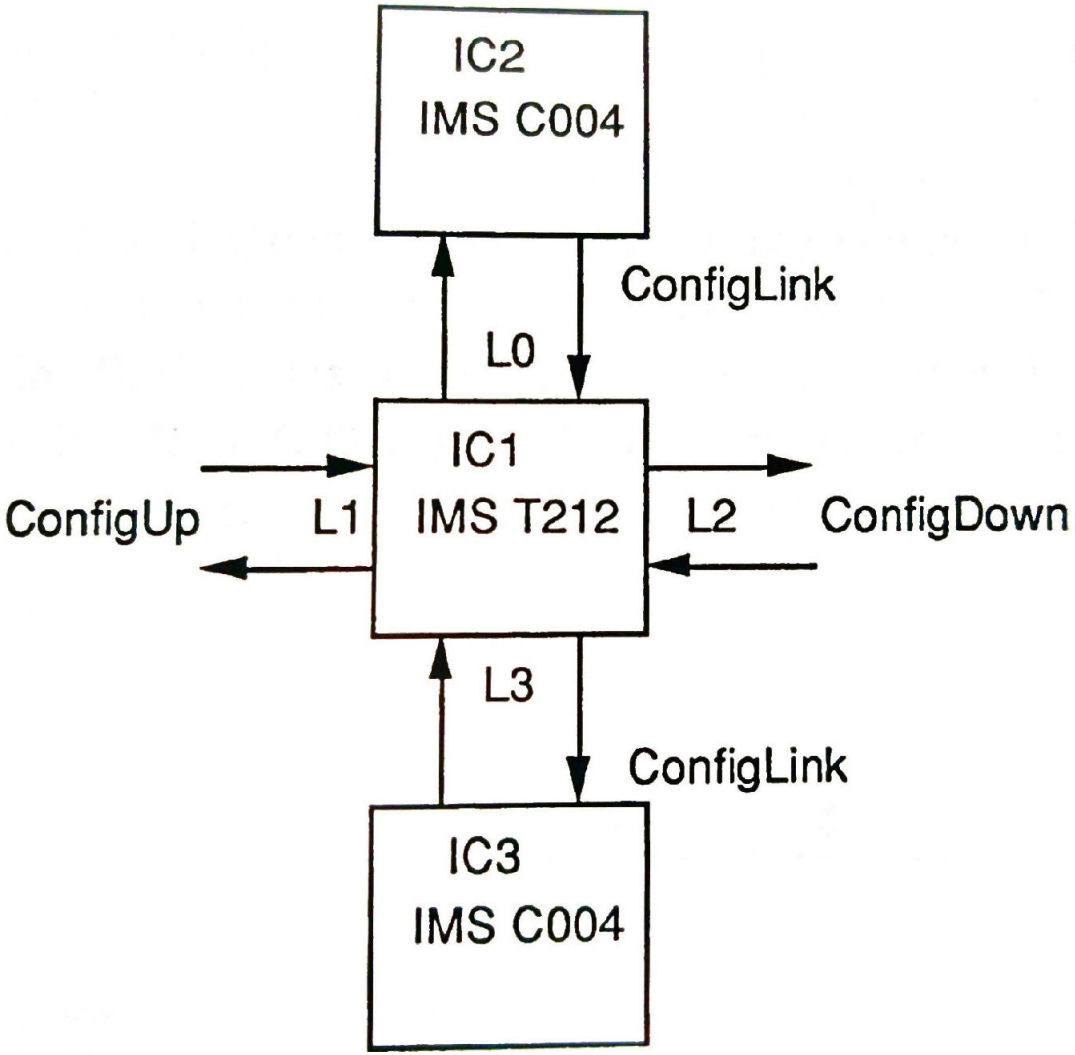



B012
T800
T800
T800
T800
T800
T800
T800
T800
B012
T800
T800
T800
T800
T800
T800
T800
T800
B014
HOST

ANFITRION
L1
L2
Subsystem
IMS B012
PipeHead
ConfigUp
ConfigDown
PipeTail
Up
Down
IMS B012
PipeTail
ConfigUp
Up

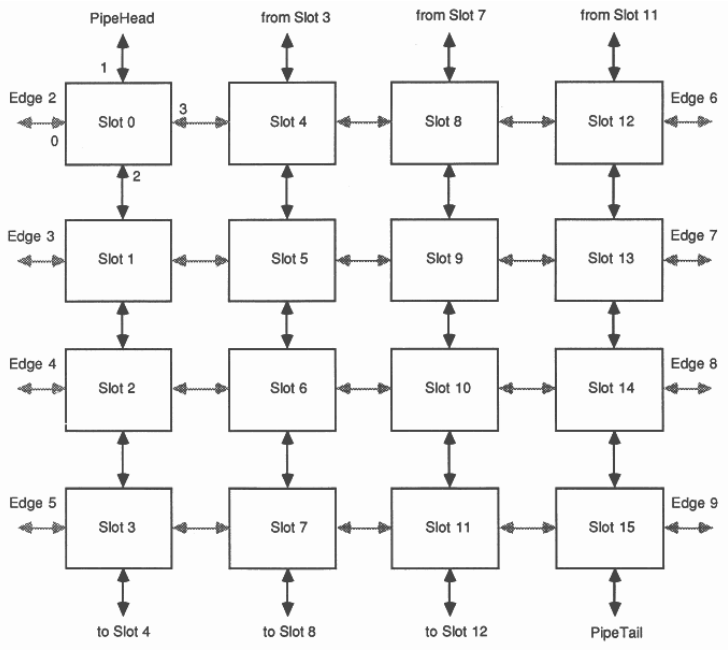

P2
to C004s
to C004s
K1
0
1
P2
Slot0
2
Slot1
Slot2
Slot3
K1
3
K1
Slot4
Slot5
Slot6
Slot7
K1
K1
Slot8
Slot9
Slot10
Slot11
K1
K1
Slot12
Slot13
Slot14
Slot15
P2
P2
K1


Edge Connector Links
IMS C004 (IC2)
PipeHead
PipeTail
LinkOut3
LinkIn3
LinkIn1
LinkOut2
LinkOut1
LinkIn2
Slot 0
Slot 15
LinkIn0
LinkOut0
IMS C004 (IC3)
Edge Connector Links


IC2
IMS C004
ConfigLink
L0
IC1
IMS T212
ConfigUp
L1
L2
ConfigDown
L3
ConfigLink
IC3
IMS C004

Configuración malla de una B012 en HL1 (MMS)

SOFTWIRE SLOT 0,0 TO EDGE 2 SLOT 0,3 TO SLOT 4,0 SLOT 4,3 TO SLOT 8,0 SLOT 8,3 T
O SLOT 12,0 SLOT 12,3 TO EDGE 6 SLOT 1,0 TO EDGE 3 SLOT 1,3 TO SLOT 5,0 SLOT 5,3 T
O SLOT 9,0 SLOT 9,3 TO SLOT 13,0 SLOT 13,3 TO EDGE 7 SLOT 2,0 TO EDGE 4 SLOT 2,3 T
O SLOT 6,0 SLOT 6,3 TO SLOT 10,0 SLOT 10,3 TO SLOT 14,0 SLOT 14,3 TO EDGE 8 SLOT 3,
0 TO EDGE 5 SLOT 3,3 TO SLOT 7,0 SLOT 7,3 TO SLOT 11,0 SLOT 11,3 TO SLOT 15,0 SLOT
15,3 To EDGE 9 END



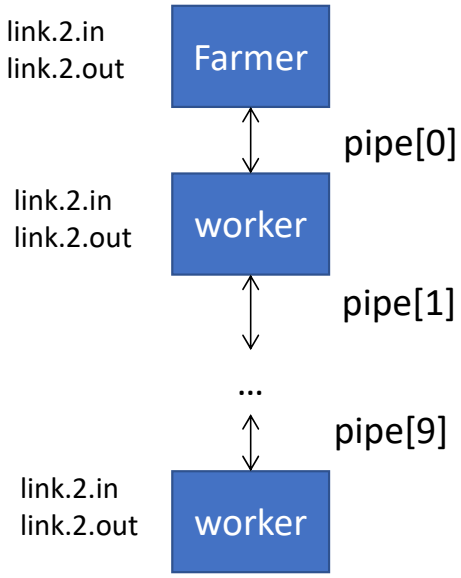
```
--{{{  DEF B014
DEF B014
 --{{{  SIZES
 SIZES
   T2   1
   C4   2
   SLOT 8
   EDGE 27
 END
 --}}}
 --{{{  T2CHAIN
 T2CHAIN
   T2 0, LINK 0 C4 0
   T2 0, LINK 3 C4 1
 END
 --}}}
 --{{{  HARDWIRE
 HARDWIRE
   --{{{  TRAMS TO C004S
   --{{{  TRAM 0
   C4 1, LINK 1 TO SLOT 0, LINK 2
   C4 0, LINK 1 TO SLOT 0, LINK 3
   --}}}
   --{{{  TRAM 1
   C4 0, LINK 2 TO SLOT 1, LINK 0
   C4 1, LINK 2 TO SLOT 1, LINK 1
   C4 1, LINK 3 TO SLOT 1, LINK 2
   C4 0, LINK 3 TO SLOT 1, LINK 3
   --}}}
   --{{{  TRAM 2
```

```
   --{{{  TRAM 6
   C4 0, LINK 12 TO SLOT 6, LINK 0
   C4 1, LINK 12 TO SLOT 6, LINK 1
   C4 1, LINK 13 TO SLOT 6, LINK 2
   C4 0, LINK 13 TO SLOT 6, LINK 3
   --}}}
   --{{{  TRAM 7
   C4 0, LINK 14 TO SLOT 7, LINK 0
   C4 1, LINK 14 TO SLOT 7, LINK 1
   C4 1, LINK 15 TO SLOT 7, LINK 2
   C4 0, LINK 15 TO SLOT 7, LINK 3
   --}}}
 --}}}

 --{{{  EDGES 0..7 (P2) TO C004 A (0)
 C4 0, LINK 16 TO EDGE 0
 C4 0, LINK 17 TO EDGE 1
 C4 0, LINK 18 TO EDGE 2
 C4 0, LINK 19 TO EDGE 3
 C4 0, LINK 20 TO EDGE 4
 C4 0, LINK 21 TO EDGE 5
 C4 0, LINK 22 TO EDGE 6
 C4 0, LINK 23 TO EDGE 7
 --}}}
 --{{{  EDGES 8..15 (P4) TO C004 B (1)
 C4 1, LINK 16 TO EDGE 8
```

```
DEF B1

SIZES
  T2     1
  C4     2
  SLOT   16
  EDGE   32
END

T2CHAIN
  T2 0, LINK 0 C4 0
  T2 0, LINK 3 C4 1
END
HARDWIRE
  SLOT 0, LINK 2 TO SLOT 1, LINK 1
  SLOT 1, LINK 2 TO SLOT 2, LINK 1
  SLOT 2, LINK 2 TO SLOT 3, LINK 1
  SLOT 3, LINK 2 TO SLOT 8, LINK 1

  SLOT 11, LINK 1 TO SLOT 10, LINK 2
  SLOT 10, LINK 1 TO SLOT 9, LINK 2
```

Asignación procesos - procesadores en Occam

- PLACED PAR
- PROCESSOR *number type*
- PLACE *channel* AT *link.number*




```
CHAN OF ANY pipe
VAL link.2.out IS 2:
VAL link.1.in IS 5:

PLACED PAR

        PROCESSOR 1 T8
                PLACE pipe[0] AT link.2.out:
                farmer(pipe[0])

        PLACED PAR i=1 FOR 9

                PROCESSOR i T8
                VAL siguiente IS i:
                VAL anterior IS i-1:
                PLACE pipe[siguiente] AT link.2.out :
                PLACE pipe[anterior] AT link.1.in :

                worker(i, pipe[siguiente),pipe[anterior]
```

Rutina de trabajo con el TDS

Todo en un transputer

```
icc foo -t8
ilink foo.tco -t8 -f starup.lnk
icollect foo.lku -t
iserver -sb foo.btl
```

En red de transputers

```
icc foo -t8
ilink foo.tco -t8 -f starup.lnk
icconf foo.cfs
icollect foo.lku -t
iserver -sb foo.btl
```

H1 – T9000 (1993)

- 2 M transistors
- Pipelined superscalar (5 stages)
- Communications processor
- 16 KB fully associative cache
- 36 MIPS @50 MHz

– "Best host for a T9000?"

–"An overhead projector!"