McGill

ECSE 421 Lecture 10: More Processing Elements

ESD Chapter 3

Last Time

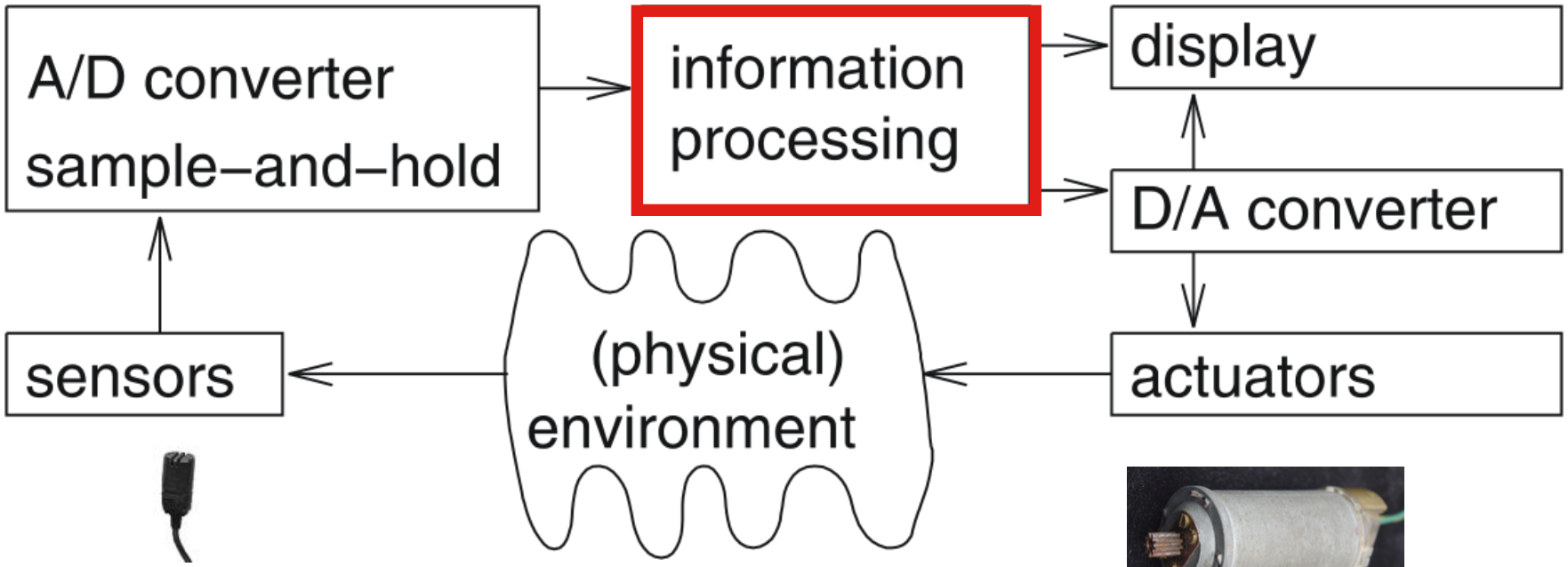
- Embedded systems employ hardware in a loop
- Information processing
  - Importance of energy efficiency
  - Special purpose HW (ASICs) are very expensive
- Key Requirements for Processors
  - Energy efficiency of processors
  - Code-size efficiency
  - Run-time efficiency

Where Are We?

W	D	Date		Topic	ESD	PES	Out	In	Notes
1	T	12-Jan-2016	L01	Introduction to Embedded System Design	1.1-1.4				
	R	14-Jan-2016		Introduction to Embedded System Design	1.1-1.4				
2	T	19-Jan-2016	L02	Specifying Requirements / MoCs / MSC	2.1-2.3				
	R	21-Jan-2016	L03	CFSMs	2.4				
3	T	26-Jan-2016	L04	Data Flow Modeling	2.5	3.1-5,7	LA1		
	R	28-Jan-2016	L05	Petri Nets	2.6				Thru Slide 21
4	T	2-Feb-2016	L06	Discrete Event Models	2.7	4			G: Zaid Al-bayati
	R	4-Feb-2016	L07	DES / Von Neumann Model of Computation	2.8-2.10	5	LA2	LA1	
5	T	9-Feb-2016	L08	Sensors	3.1-3.2	7.3,12.1-6			
	R	11-Feb-2016	L09	Processing Elements	3.3	12.6-12			
6	T	16-Feb-2016	L10	More Processing Elements / FPGAs			LA3	LA2	
	R	18-Feb-2016	L11	Memories, Communication, Output	3.4-3.6				
7	T	23-Feb-2016	L12	Embedded Operating Systems	4.1				
	R	25-Feb-2016		***Midterm exam: in-class, closed book***			P	LA3	Chapters 1-3
	T	1-Mar-2016		***No class***					Winter break
	R	3-Mar-2016		***No class***					Winter break
8	T	8-Mar-2016	L13	Middleware	4.4-4.5				
	R	10-Mar-2016	L14	Performance Evaluation	5.1-5.2				
9	T	15-Mar-2016	L15	More Evaluation and Validation	5.3-5.8				
	R	17-Mar-2016	L16	Introduction to Scheduling	6.1-6.2.2				
10	T	22-Mar-2016	L17	Scheduling Aperiodic Tasks	6.2.3-6.2.4				
	R	24-Mar-2016	L18	Scheduling Periodic Tasks	6.2.5-6.2.6				

Today

- Embedded system hardware is frequently used in a loop ("hardware in a loop"):



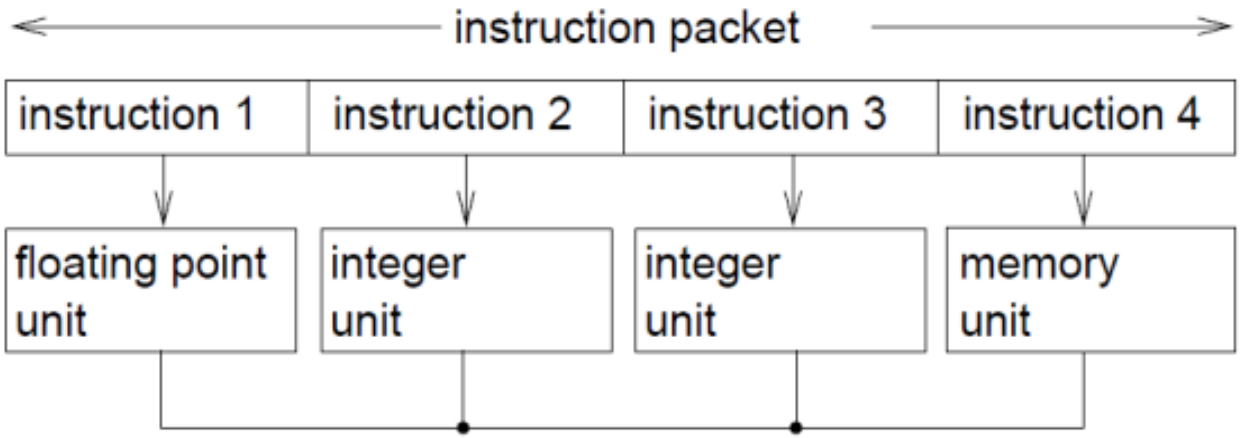
☞ cyber-physical systems

Recall: Three Key Requirements

- Embedded processing elements must achieve
  - Energy efficiency
  - Code-size efficiency
  - Run-time efficiency
- Often, this is done with accomplished with special architectures or architectural features
  - Digital signal processors
  - VLIW processors
  - Domain-specific processors
  - Application-specific processors

VLIW Processors

- Very Long Instruction Word (VLIW) Computing
  - Instructions grouped in packets
  - Instructions in a packet are executed in parallel
- Packet locations are associated with fixed functional units
  - Performance limited by compiler's ability to fill slots

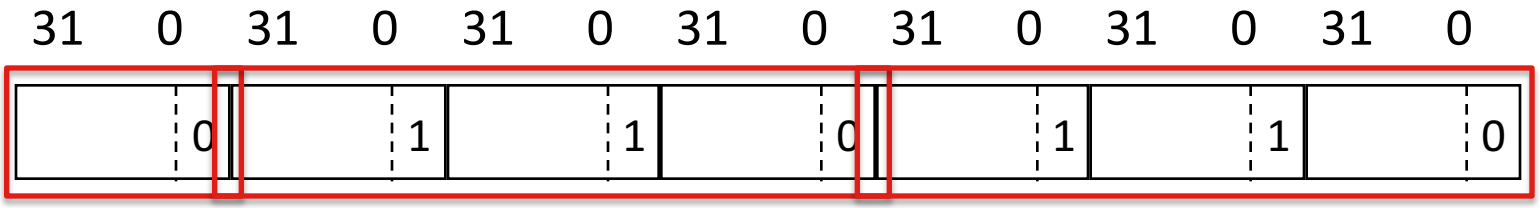


VLIW vs. RISC

- RISC: hardware finds independent instructions dynamically at *run-time*
  - Hardware overhead in area, energy, etc.
- VLIW: compiler finds independent instructions to execute in packets at *compile-time*
  - Eliminates the above overhead, but …
  - Not all parallelism can be exposed at compile-time (e.g., memory disambiguation)
- EPIC: *explicitly parallel instruction set computers*
  - Parallelism is identified by the compiler, but …
  - Parallelism is not limited to instruction packets

EPIC Example: TMS 320C6xx

- 1 bit per instruction encodes end of parallel exec.



Instr. A | Instr. B | Instr. C | Instr. D | Instr. E | Instr. F | Instr. G

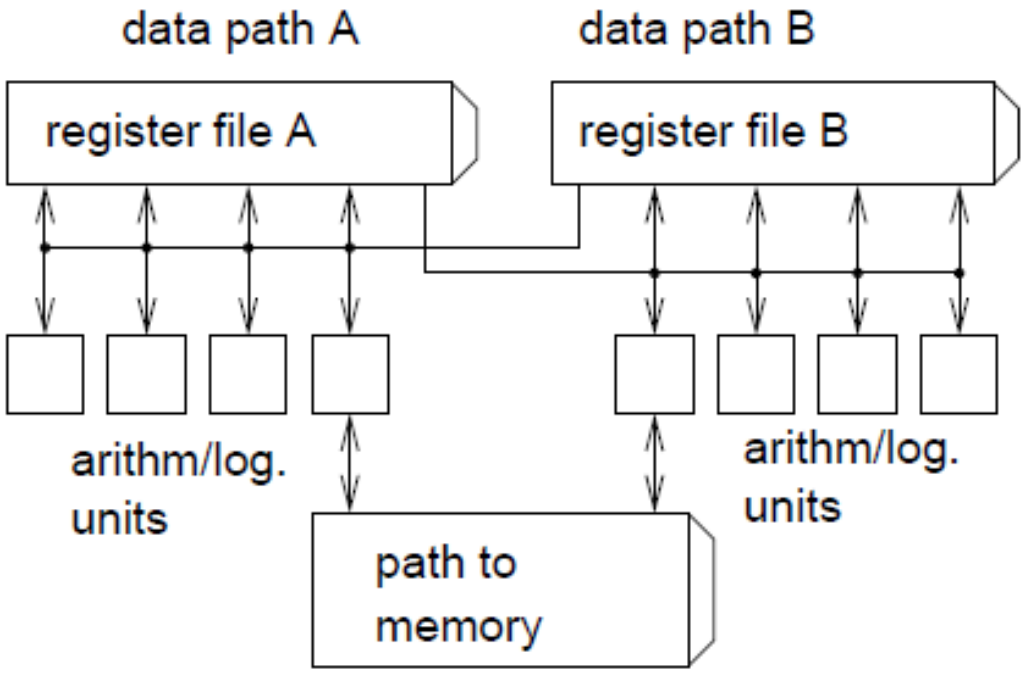
Cycle	Instruction		
1	A		
2	B	C	D
3	E	F	G

Instructions B, C and D use disjoint functional units, cross paths and other data path resources. The same is also true for E, F and G.

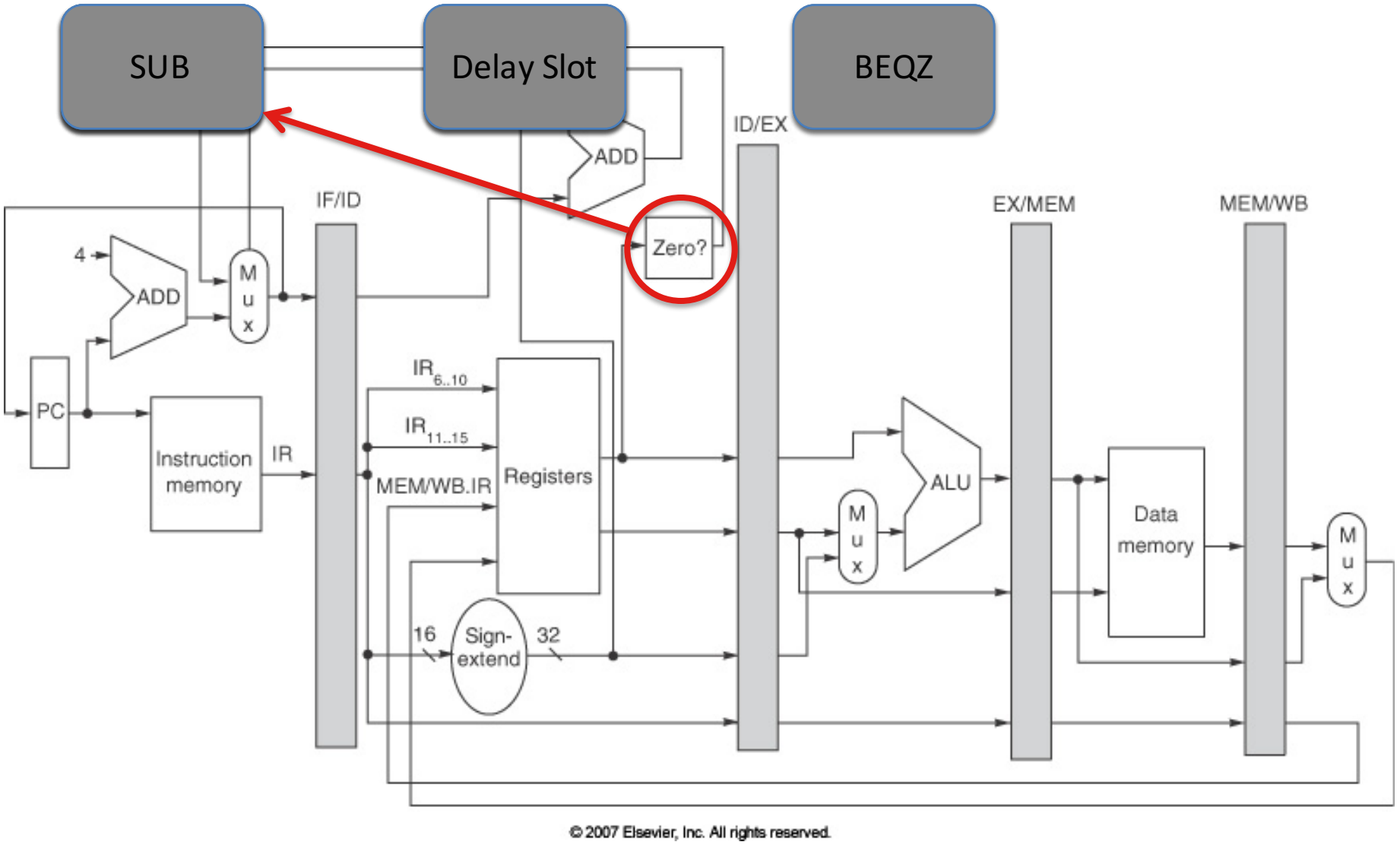
- Parallel execution cannot span several packets

Partitioned Register Files

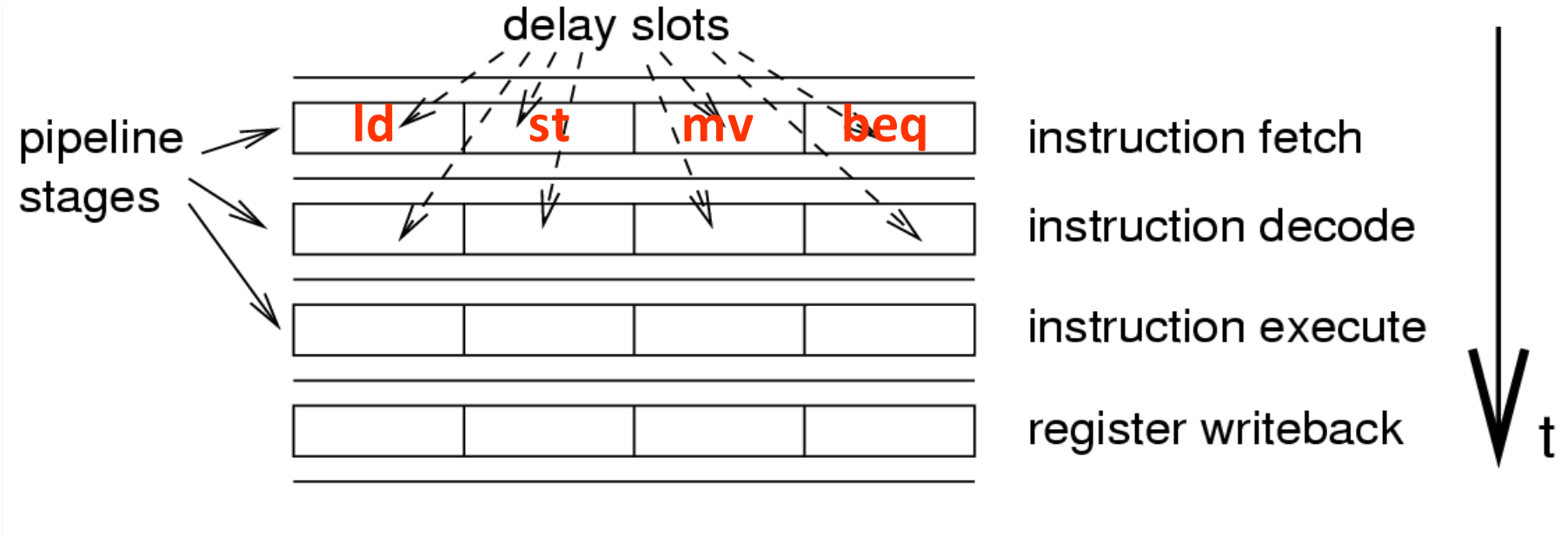
- Register files with many ports are required to deliver many operands each cycle
- Memories with many ports are expensive
- ⇒Registers are partitioned into (typically 2) sets
- *E.g.* for TI C60x:



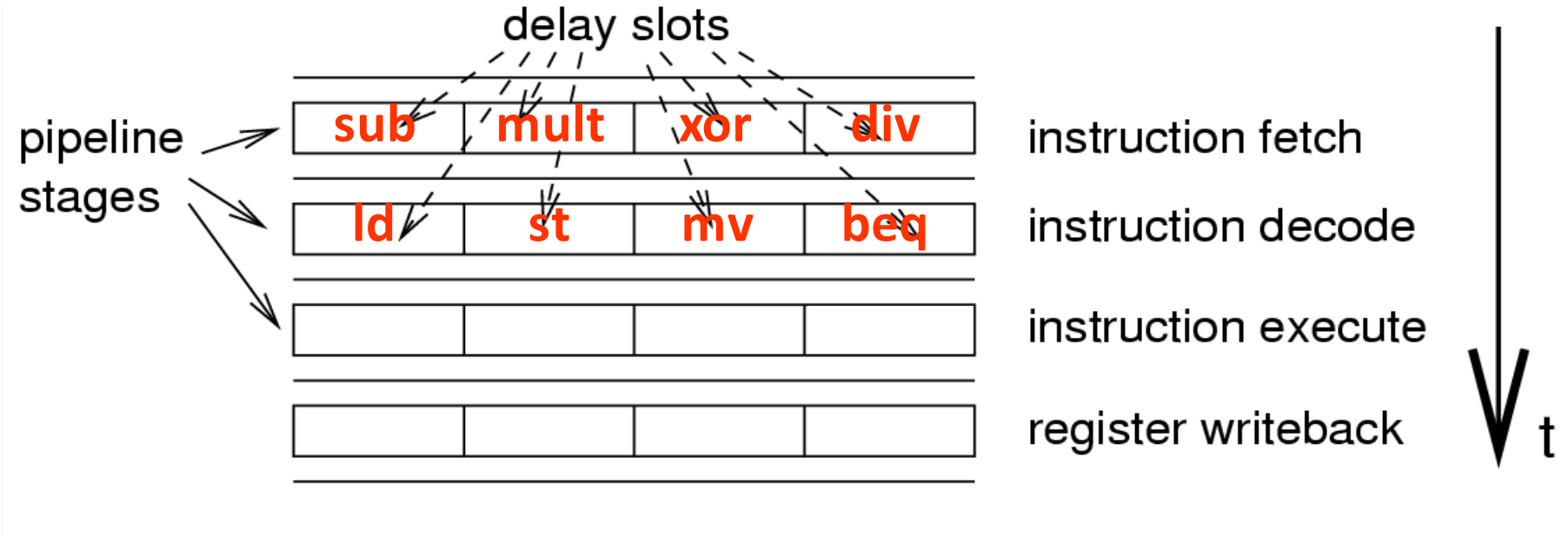
Recall: Branch Delay Slot



VLIW: Many Branch Delay Slots

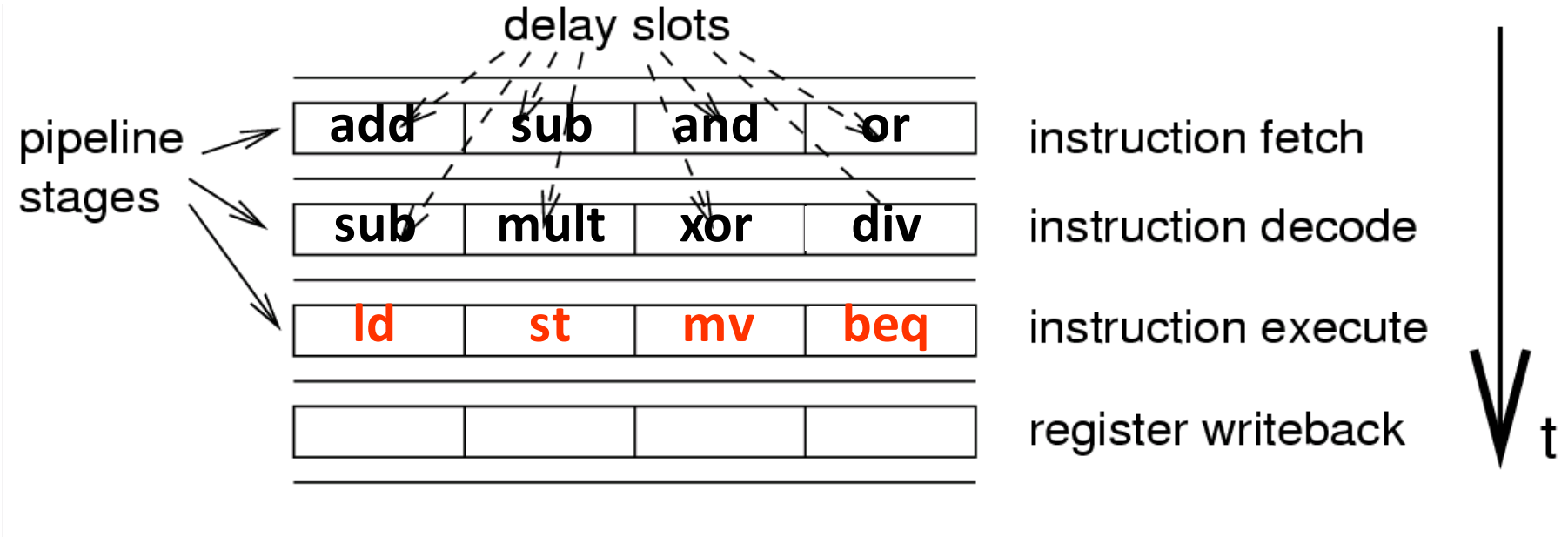


VLIW: Many Branch Delay Slots



VLIW: Many Branch Delay Slots

- Many instructions begin execution before the branch is detected
- Squashing those instructions would waste compute power



⇒ Executing those instructions is declared a feature, not a bug

⇒ Compiler must fill the "delay slots" (otherwise, waste!)

⇒ How? Avoid branches wherever possible (e.g., loop unrolling)

Predicated Instruction Execution

- Conditional Instruction "[c] I" consists of:
  - condition **c**, and
  - instruction **I**

c == true $\Rightarrow$ **I** is executed
c == false $\Rightarrow$ NOP

Predicated Execution: TI C6x

```
if (c)
{ a = x + y;
  b = x + z;
}
else
{ a = x - y;
  b = x - z;
}
```

Conditional branch

```
      [c] B L1
          NOP 5
          B L2
          NOP 4
          SUB x,y,a
      ||  SUB x,z,b
L1:       ADD x,y,a
      ||  ADD x,z,b
L2:
```

max. 12 cycles

Predicated execution

```
    [c]  ADD x,y,a
|| [c]  ADD x,z,b
|| [!c] SUB  x,y,a
|| [!c] SUB  x,z,b
```

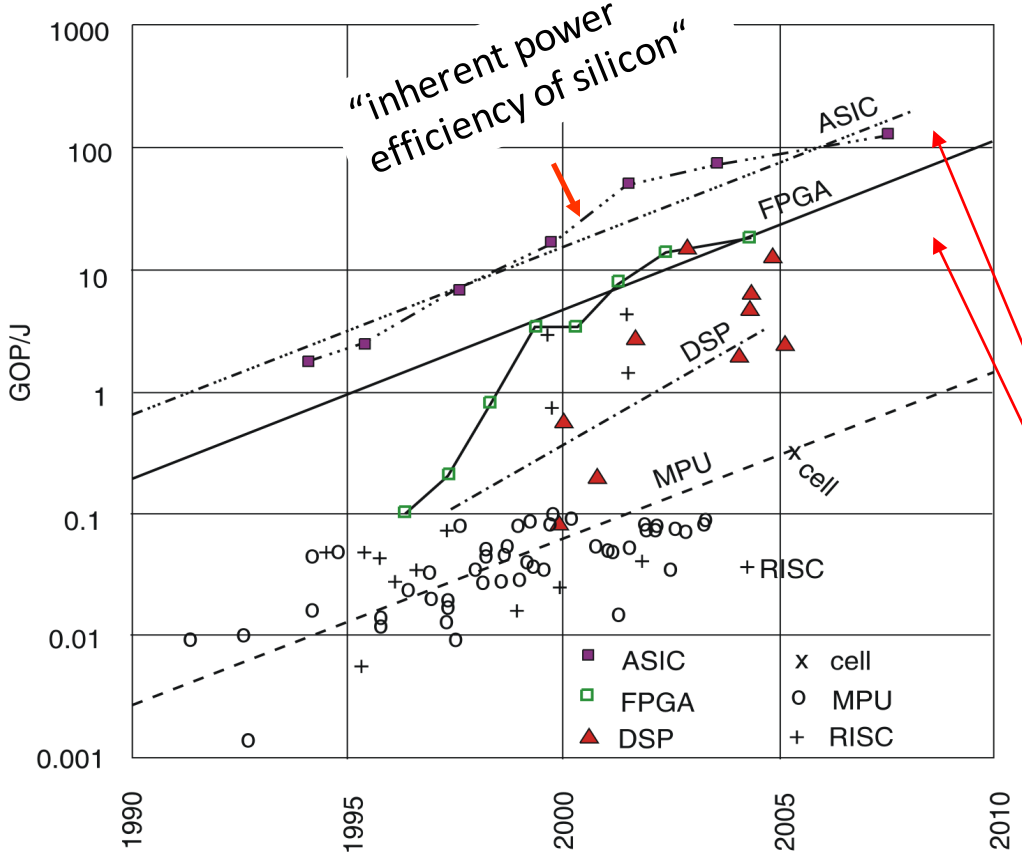
1 cycle

Microcontrollers: MHS 80C51

- 8-bit CPU optimized for control applications
- Extensive Boolean processing capabilities
- 64 k Program Memory address space
- 64 k Data Memory address space
- 4 k bytes of on chip Program Memory
- 128 bytes of on chip data RAM
- 32 bi-directional and individually addressable I/O lines
- Two 16-bit timers/counters
- Full duplex UART
- 6 sources/5-vector interrupt structure with 2 priority levels
- On chip clock oscillators
- Very popular CPU with many different variations

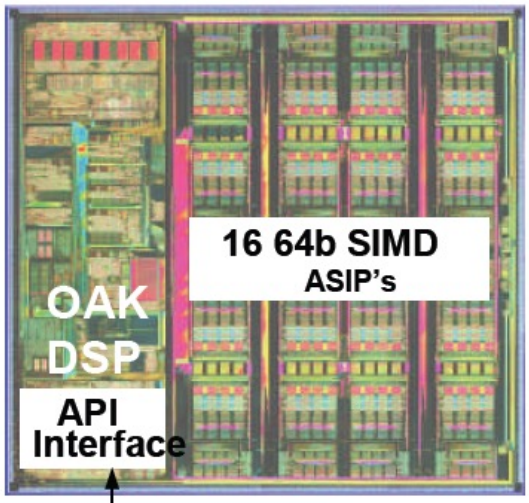
Features for Embedded Systems

Domain- and Application-Specific PEs



© Hugo De Man: From the Heaven of Software to the Hell of Nanoscale Physics: An Industry in Transition, *Keynote Slides*, ACACES, 2007

VIP for car mirrors Infineon

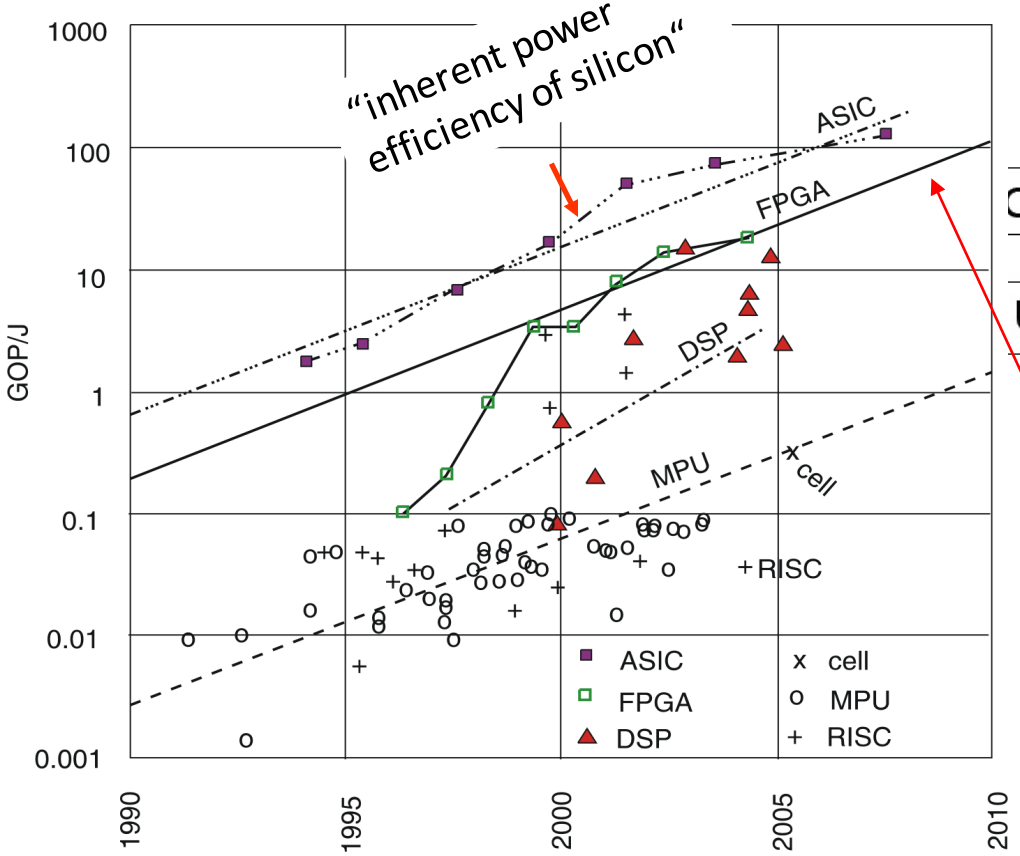


Hd Compiler ← VPL C

200MHz , 0.76 Watt
100Gops @ 8b
25Gops @ 32b

Close to power efficiency of silicon

Domain- and Application-Specific PEs

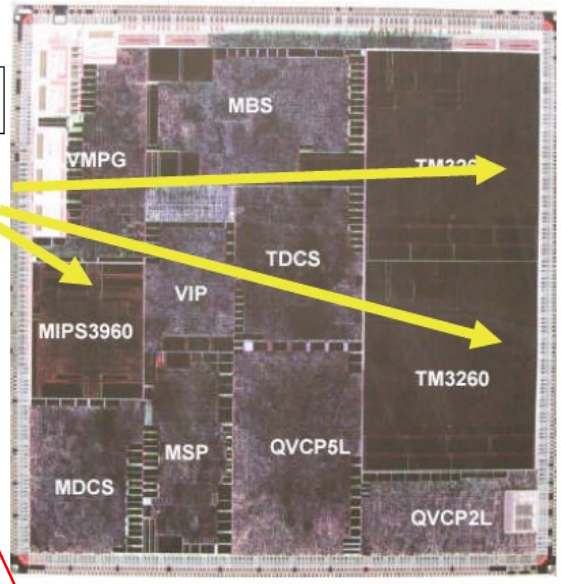


© Hugo De Man: From the Heaven of Software to the Hell of Nanoscale Physics: An Industry in Transition, *Keynote Slides*, ACACES, 2007

Nexperia Digital Video Platform
NXP

C,C++

UHAPI

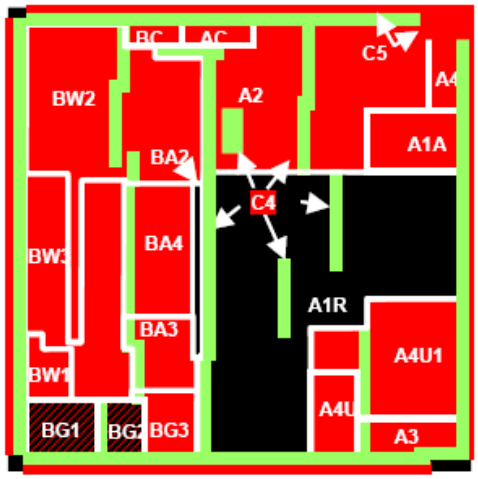


1 MIPS, 2 Trimedia
60 coproc, 250 RAM's
266MHz, 1.5 watt 100 Gops

Close to power efficiency of silicon

Multiprocessor Systems-on-a-chip

(2)Telephony (W-CDMA)



Power on

Power off

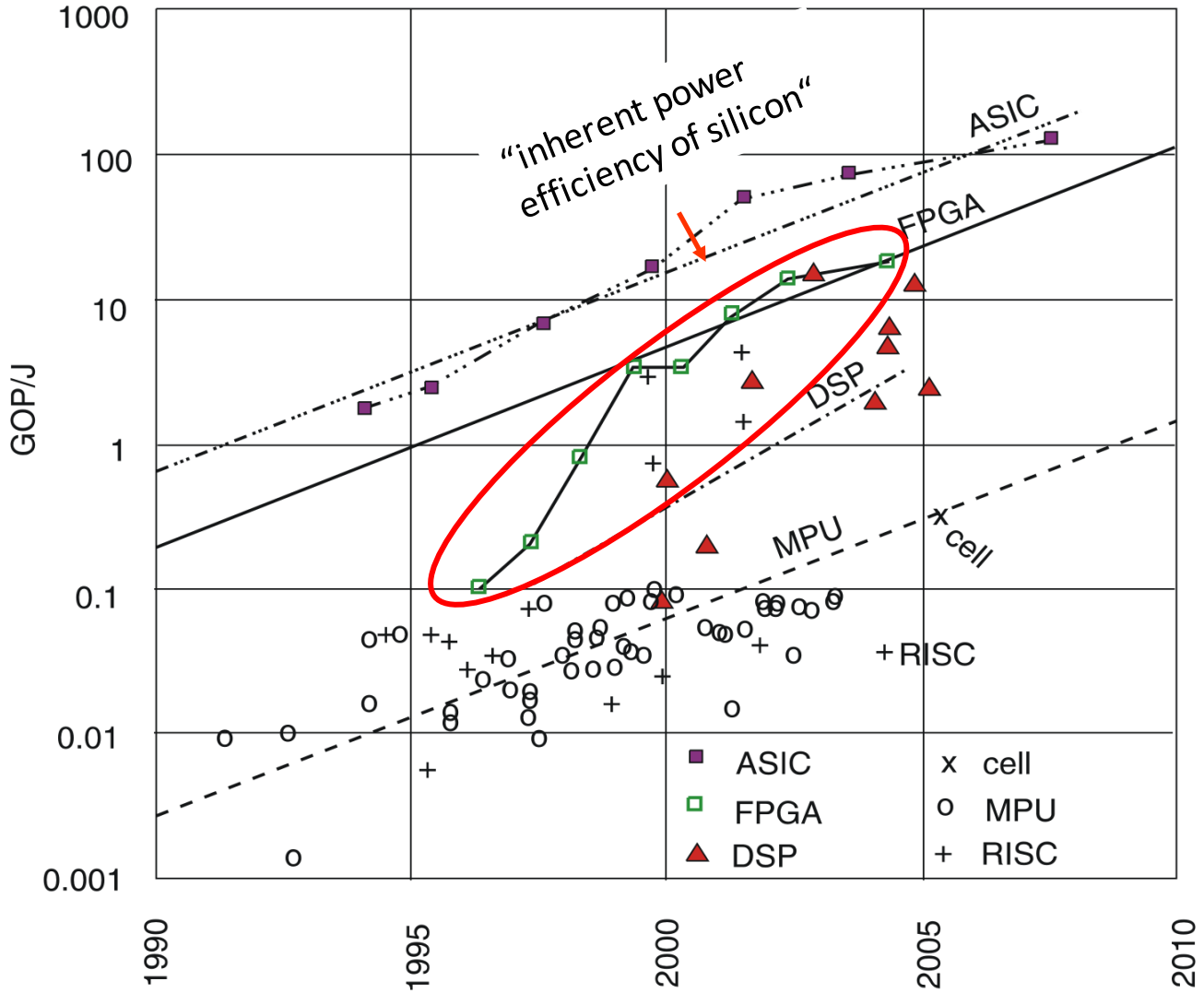
Baseband part	Control	ON
	W-CDMA	ON
	GSM	ON / OFF
Application part	System-domain	ON
	Realtime-domain	OFF
Measured Leakage Current (@ Room Temp, 1.2V)		407 μA

http://www.mpsoc-forum.org/2007/slides/Hattori.pdf

⇒ "***Dark silicon***": not all components powered at the same time

⇒ H. Esmaeilzadeh et al., "Dark Silicon and the End of Multicore Scaling," ISCA'11

Energy Efficiency of FPGAs

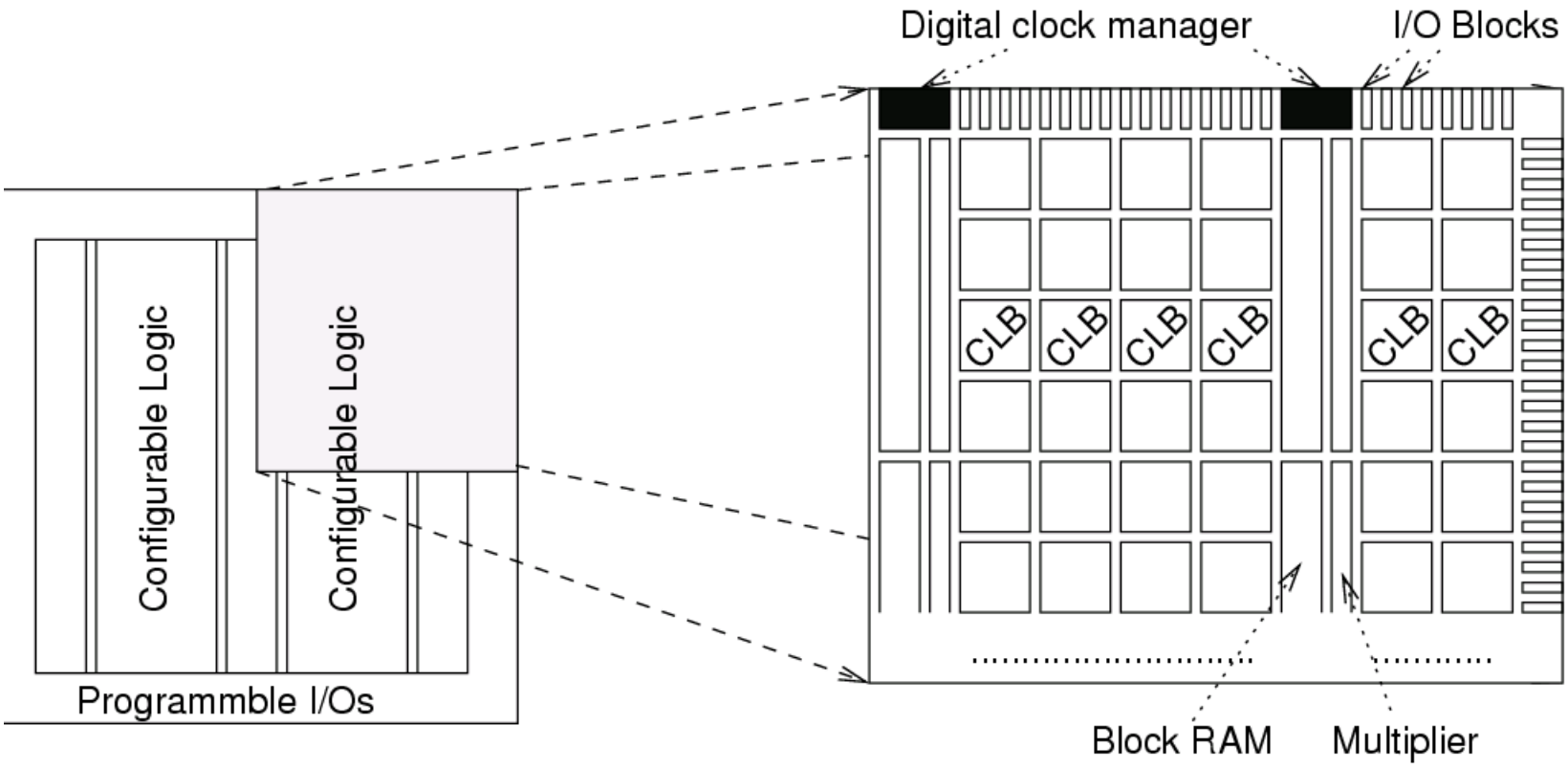


© Hugo De Man, IMEC, Philips, 2007

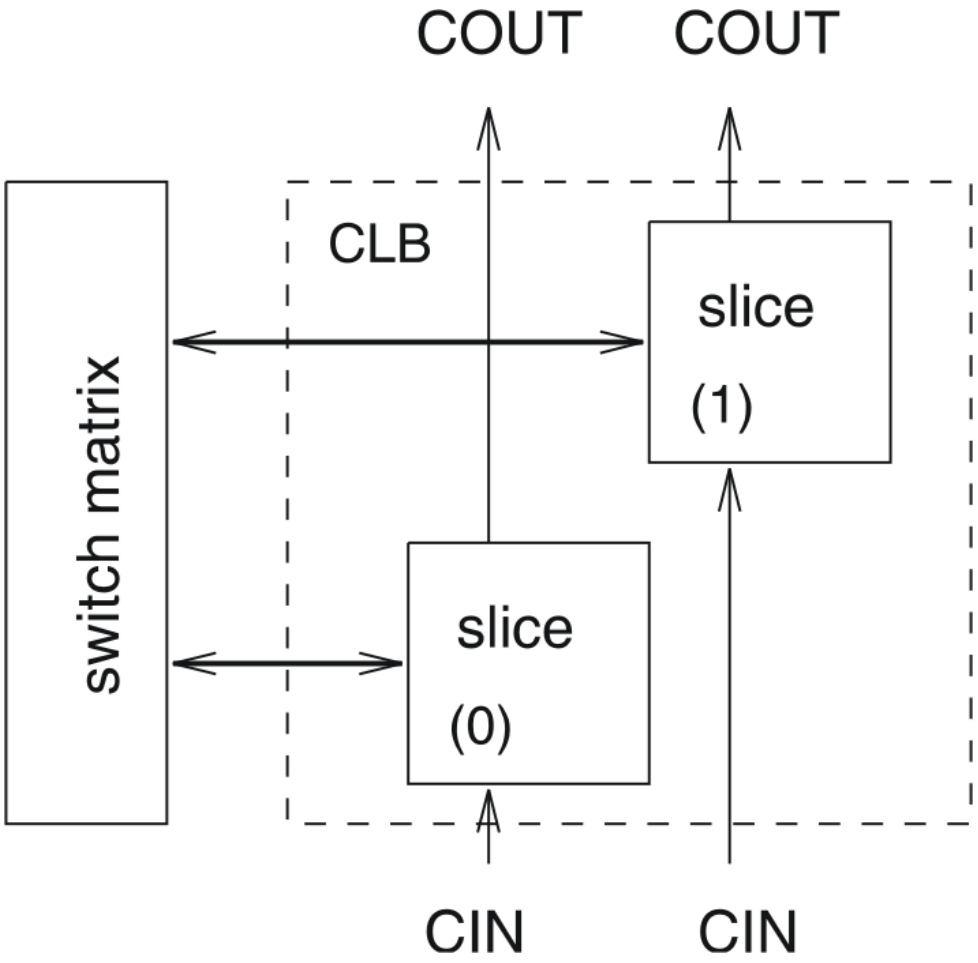
Reconfigurable Logic

- Custom HW may be too expensive
- SW may be too slow
- Combine the speed of HW with the flexibility of SW
  - HW with programmable functions and interconnect
  - Field programmable gate arrays (FPGAs)
- Applications: bit-oriented algorithms like
  - Encryption
  - Fast “object recognition” (security, medical and military)
  - Adapting mobile phones to different standards
- Very popular devices from
  - XILINX (XILINX Virtex 7 are recent devices)
  - Actel, Altera and others

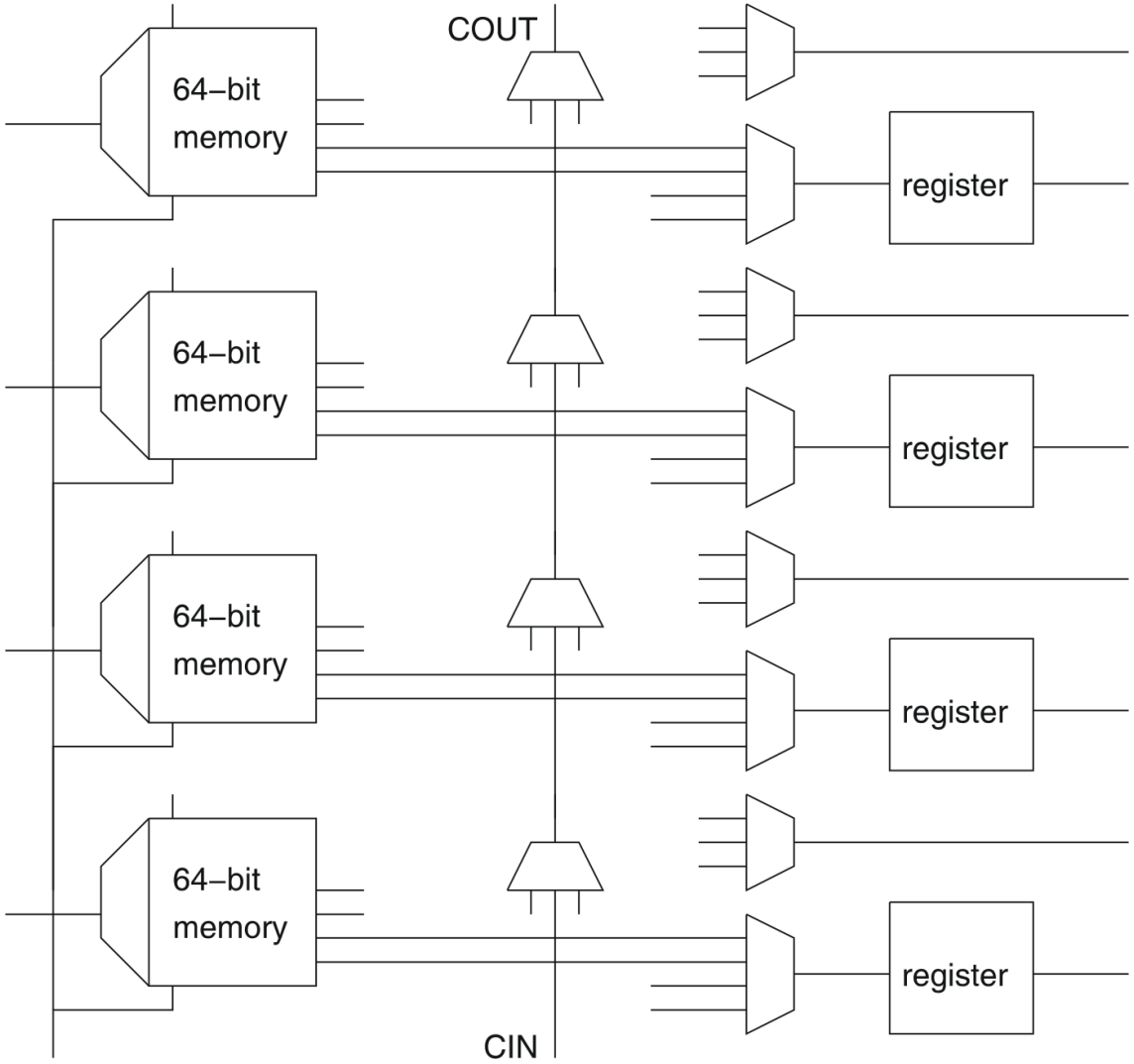
Floor-plan of VIRTEX II FPGAs



Virtex 5 Configurable Logic Block (CLB)

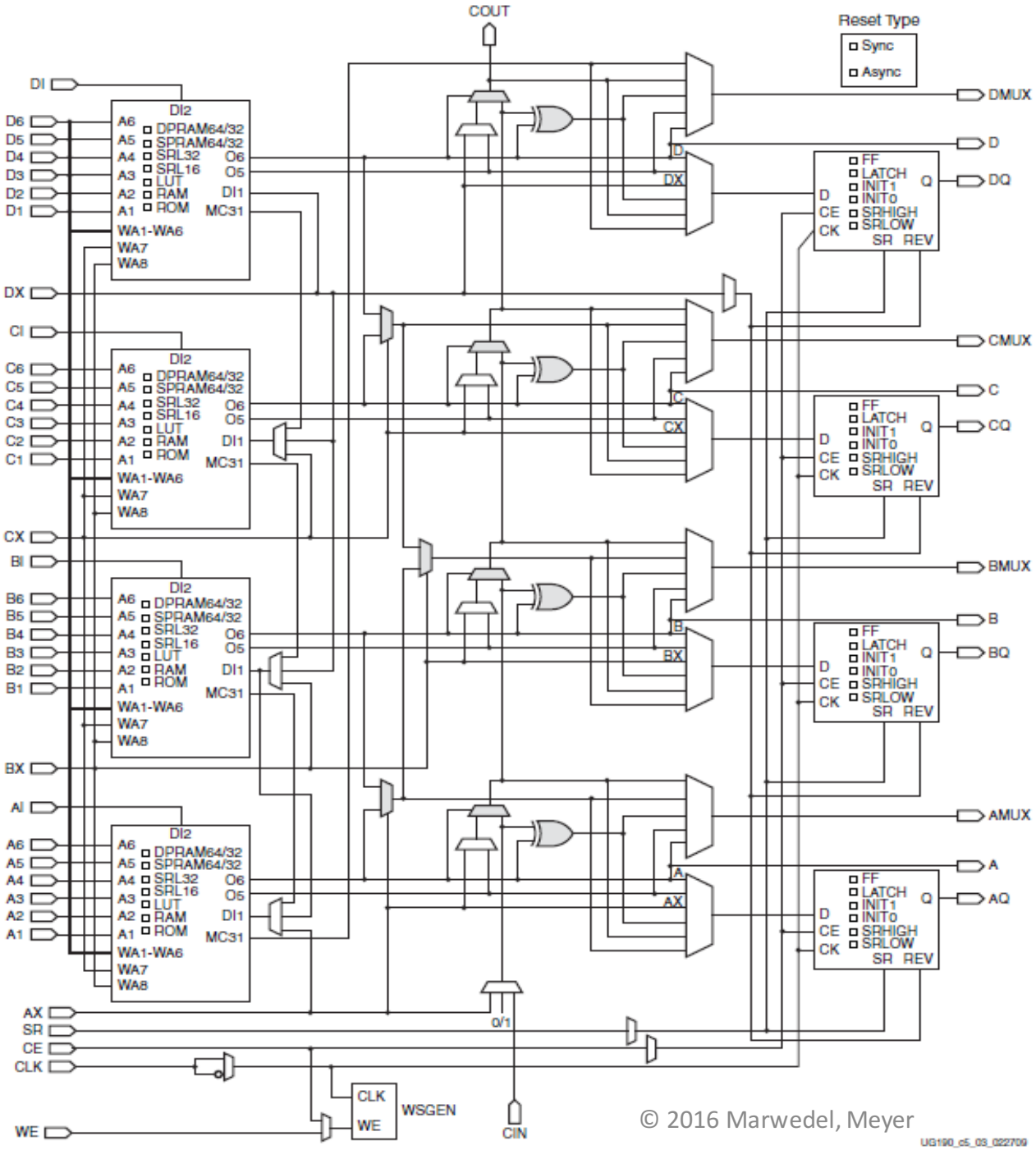


Virtex 5 Slice (Simplified)



- Memories used as look-up tables (LUTS)
- Can implement any function of ≤ 6 variables

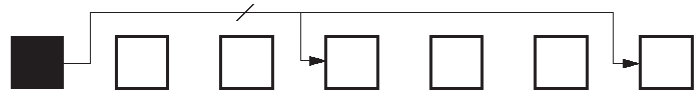
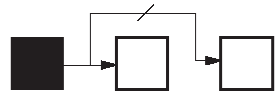
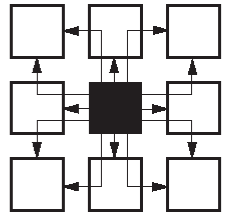
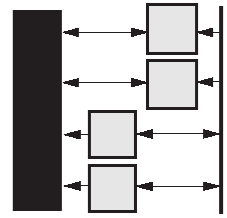
Virtex 5 SliceM



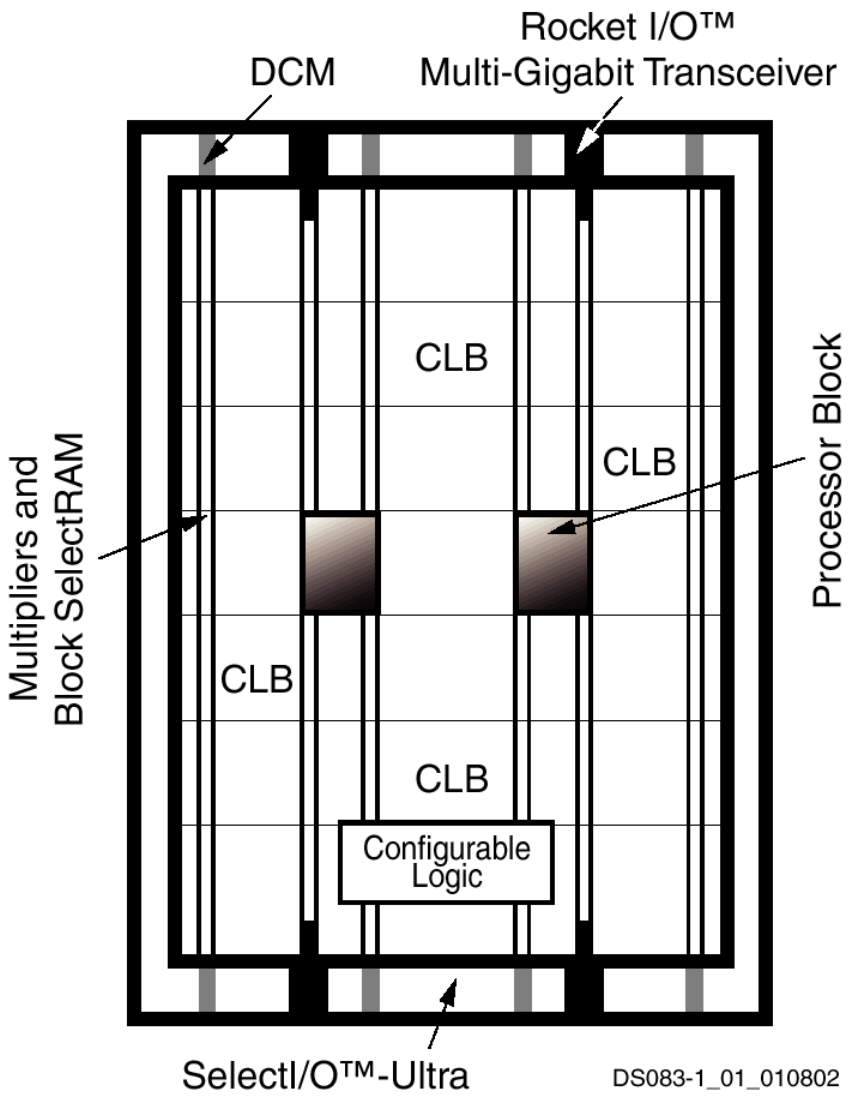
- Memories can be used for:
  - storing data,
  - shift registers

FPGA Interconnect: Virtex II

- Hierarchical Routing Resources

24 Horizontal Long Lines 24 Vertical Long Lines	
120 Horizontal Hex Lines 120 Vertical Hex Lines	
40 Horizontal Double Lines 40 Vertical Double Lines	
16 Direct Connections (total in all four directions)	
8 Fast Connects	

FPGA Hard Cores: Virtex II Pro



- Virtex II Pro devices include up to four PowerPC processor cores
  - Higher-performance than soft cores
  - IP makes it easy to interface with and between the cores
- Virtex 5 devices include up to two

[© and source: Xilinx Inc.: Virtex-II Pro™ Platform FPGAs: Functional Description, Sept. 2002, //www.xilinx.com]

Summary

- Embedded systems employ hardware in a loop
- Information processing
  - Run-time efficiency: do more in a single instruction
  - VLIW processing: parallel instructions
  - MPSoC: parallel processing elements
- Re-configurable logic
  - The flexibility of software combined with the performance and efficiency of hardware

Next Time

- Memories, Communication, and Output
  - Chapter 3.4-6