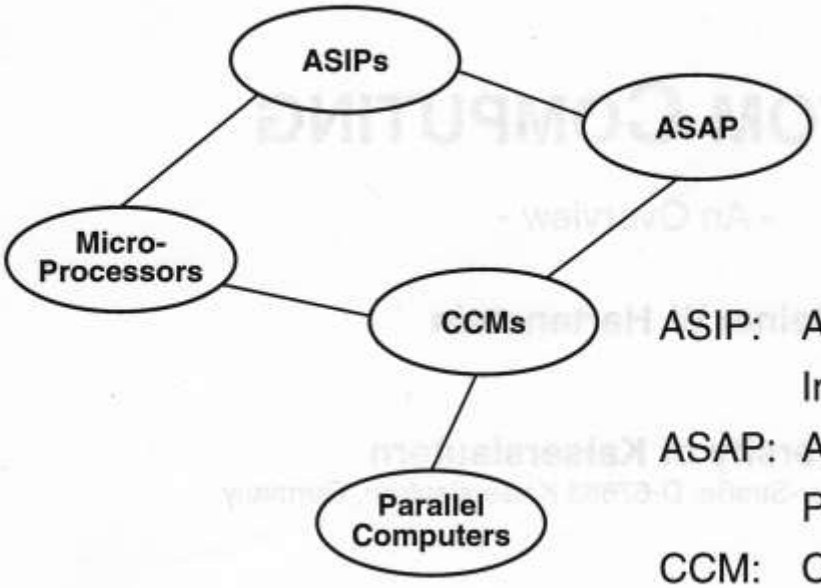
CUSTOM COMPUTING

- An Overview -

Reiner W. Hartenstein

University of Kaiserslautern
Erwin-Schrödinger-Straße, D-67663 Kaiserslautern, Germany

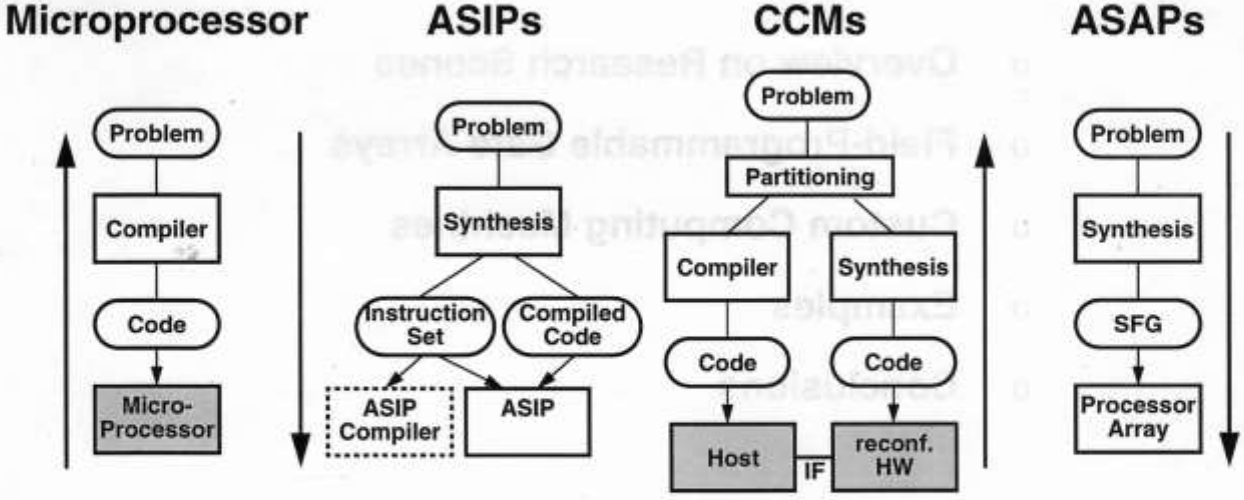
- **Overview on Research Scenes**
- **Field-Programmable Gate Arrays**
- **Custom Computing Machines**
- **Examples**
- **Conclusions**



ASIP: Application Specific Instruction Set Processor

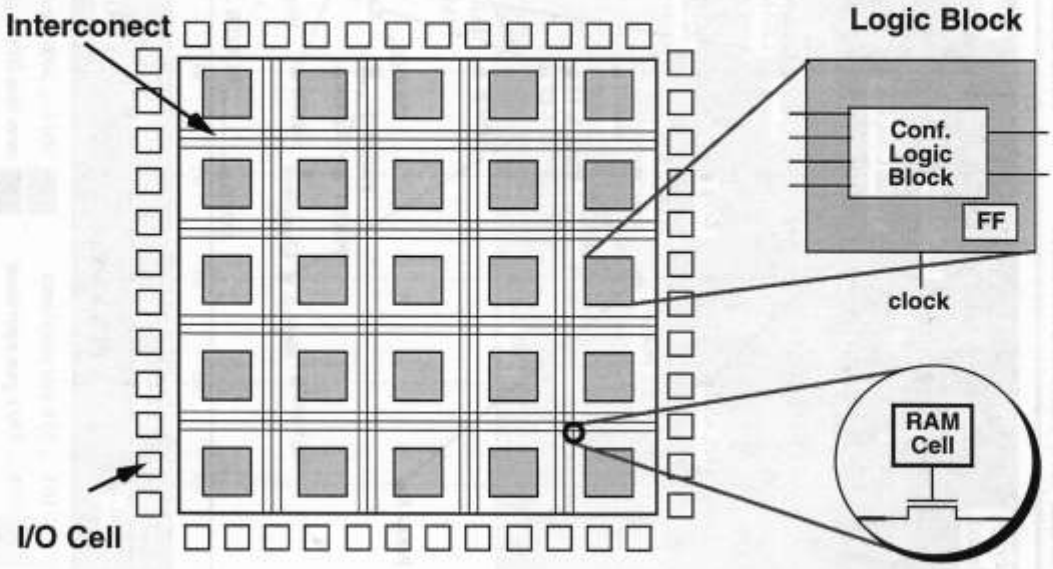
ASAP: Application Specific Array Processor

CCM: Custom Computing Machines



	Microprocessor	ASIPs	CCMs	ASAPs
TARGET:	general purpose computing	embedded controller	general purpose	special purpose
APPROACH:	bottom-up	top-down	bottom-up	top-down

already available HW planned



- **Annpolis Micro Systems, Inc.**
  - WILDFIRE Custom Configurable Computer (based on Splash 2)
- **Giga Operations Corp.**
  - Spectrum video computing engine
  - G800 VESA VL-bus PC board
- **Methalithic Systems Inc.**
  - ACE-12 Reconfigurable Compute Engine
- **Virtual Computer Corp.**
  - P-Series Virtual Computer
  - EVC-1 transformable computer

Xputer Lab Uni Kaiserslautern

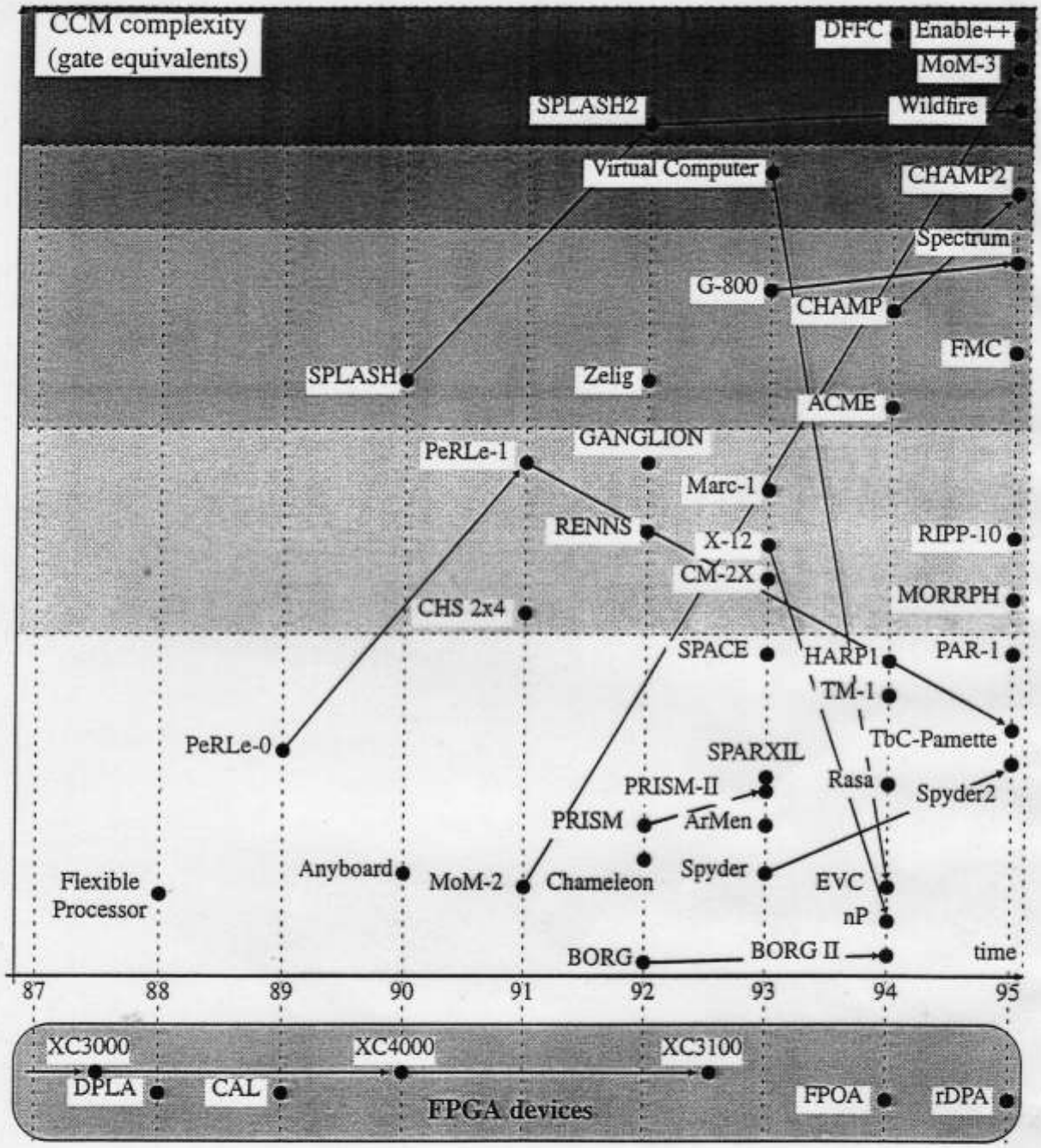
Time Scale and Size

Xputer Lab Uni Kaiserslautern

Memory Access

Xputer Lab Uni Kaiserslautern

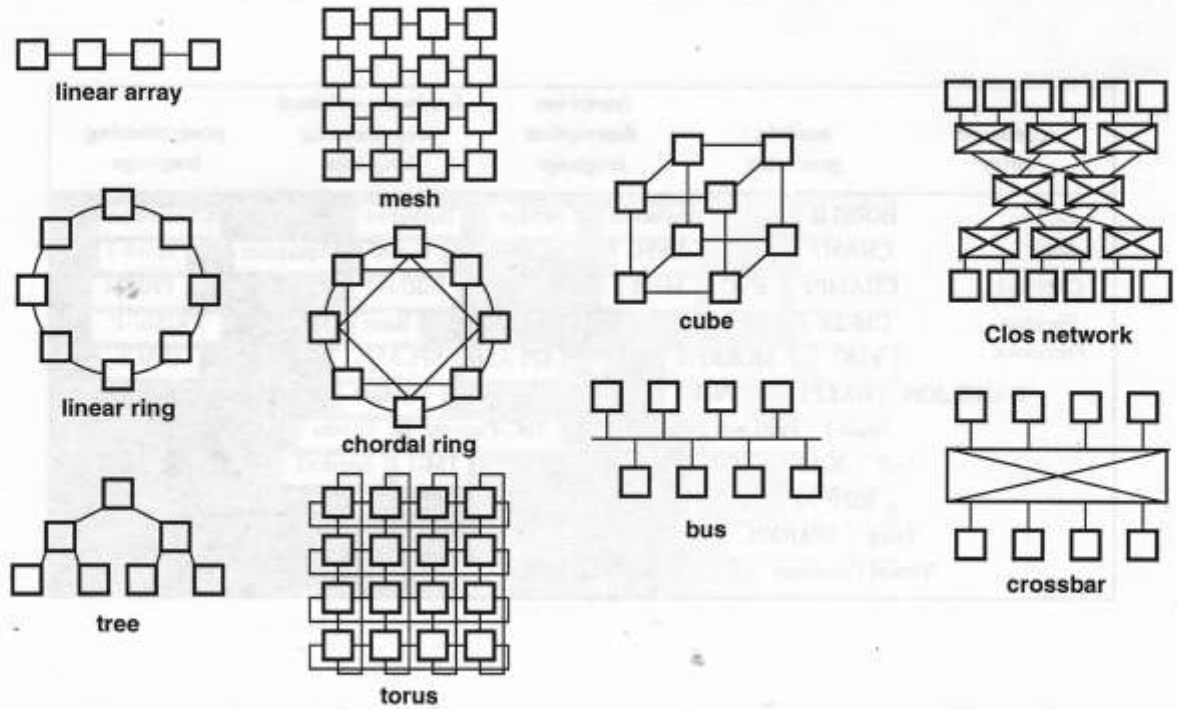
Time Scale and Size



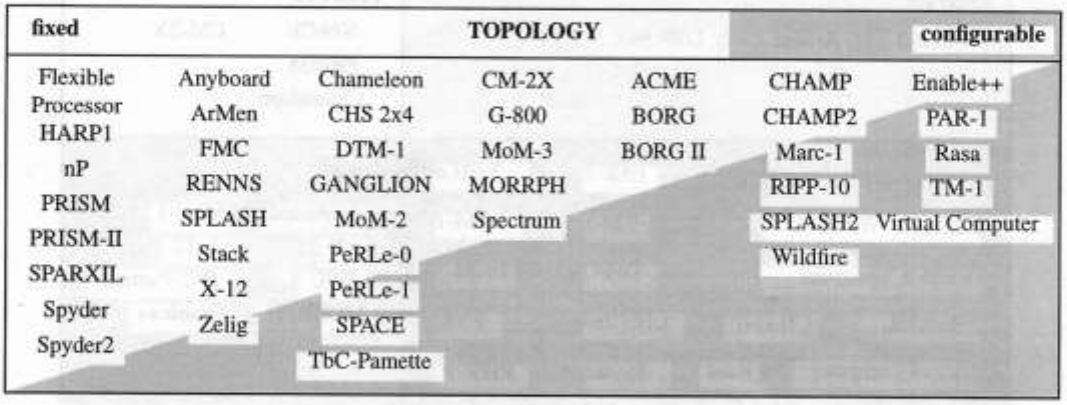
Legend:

- 10 k ... 50 k gate equivalents
- 50 k ... 150 k gate equivalents
- 150 k ... 500 k gate equivalents
- 500 k ... 2000 k gate equivalents
- more than 2000 k gate equivalents
- availability of FPGA technology

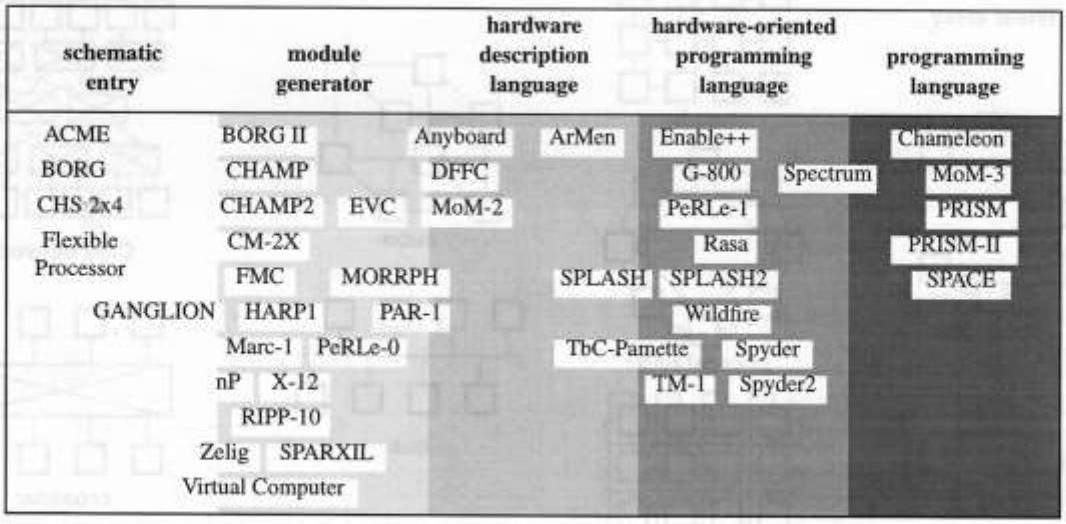
shared				remote	
MoM-2	ArMen	DSP-56X		SPACE	CM-2X
				PRISM	
				Chameleon	
shared and local				**local**	
SPLASH (Wildfire)		PAR-1	PeRLe	Anyboard	Marc-1
G-800 & Spectrum		TM-1	DTM-1	nP	X-12
SPARXIL	CHAMP	DFFC	MoM-3	EVC	TbC-Pamette
Virtual Computer	Rasa	CHS 2x4	FMC	MORRPH	Enable++
		Spyder	RIPP-10		



Flexibility of the Interconnect



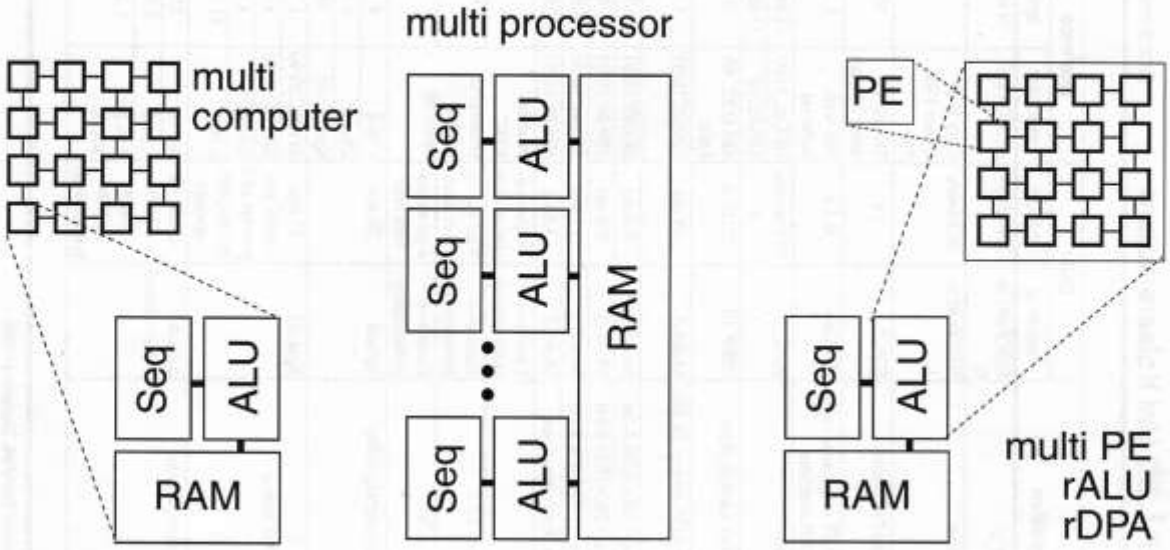
Spectrum of Programmability



algorithm	CCM		reference		acceleration factor
	machine	speed	machine	speed	
RSA cryptography	DECPeRLe-0	200 kbit/s	AT&T VLSI	19 kbit/s	10
long multiplication	DECPeRLe-1	66 Mbit/s	Cray II or Cyber 170/750		16
linear convolution (16-point)	EVC1	5 s	SPARC10 using Matlab	95 s	10
insertion sort, 1677 sequences each containing 39 7-bit numbers	CHS2x4	0.1 s	486 with turbo on	1.2 s	12
full viterbi search	Rasa	853 words/s	DECst. 3100	12.98 w/s	66
			SPARC10	35.02 w/s	24
template-based ray-casting algorithm	RJPP-10	0.166 s	486 DX2, 66 MHz	20 s	120
JPEG, 704x576 RGB image, 24 bits per pixel	MoM-3	64 ms	SPARC10/51	1.51	24
3x3 2d FIR filter, 1280x1024 8-bit	MoM-3	5.8 ms	SPARC10/51	0.71 s	122
5x5 2d FIR filter, 1280x1024 8-bit	MoM-3	9.6 ms	SPARC10/51	1.87 s	180
7x7 2d FIR filter, 1280x1024 8-bit	MoM-3	15 ms	SPARC10/51	3.58 s	213
Ising 128 lattice, 1000 iterations	MoM-3	3.85 s	SPARC10/51	1208 s	331
Calorimeter	Enable++ (estimation)	10 µs execution time	modern RISC		430
Radiation Tracker TRT	Enable++ (estimation)	9 µs execution time	(PowerPC, Alpha,		180
Silicon Tracker SCT	Enable++ (estimation)	4 µs execution time	Sparc10		600
sequence comparison algorithm	Splash	20 ms	CM-2	4.7 s	235
			Cray 2	6.5 s	325
			Sun 3/140	48 s	2400
convolution 3 x 3, on 512 x 512 pixel image	Splash2	13 ms, coeff. are powers of 2 (65 ms, otherwise)	SPARC20/40	1.7 s	131 (26)
			SPARC10/30	3.0 s	230 (46)
			SPARC2	4.5 s	346 (69)
			i-860	51.9 ms	4 (0.8)
DNA sequence comparison	Splash2 (16 modules)	43000 M CUPS (cells updated per second)	MP-1	32 M	1344
			CM-2	5.9 M	7288
			Sun 10/30GX	1.2 M	35833

- **IEEE International Workshop on FPGAs for Custom Computing Machines** (Napa, April)
- **International Workshop on Field-Programmable Logic**
- **ACM/SIGDA International Workshop on Field-Programmable Gate Arrays** (Berkeley, February)
- **List of Custom Computing Machines**
  - Â S. Gucione, U. of Texas at Austin: guccione@ccwf.cc.utexas.edu
- **Collection of Abstracts on CCM related literature**
  - Â M. J. Wirthlin, Brigham Young U.: wirthlin@fpga.ee.byu.edu

algorithm	CCM		reference		acceleration factor
	machine	speed	machine	speed	
RSA cryptography	DECPeRLe-0	200 kbit/s	AT&T VLSI	19 kbit/s	10
long multiplication	DECPeRLe-1	66 Mbit/s	Cray II or Cyber 170/750		16
linear convolution (16-point)	EVC1	5 s	SPARC10 using Matlab	95 s	10
insertion sort, 1677 sequences each containing 39 7-bit numbers	CHS2x4	0.1 s	486 with turbo on	1.2 s	12
full viterbi search	Rasa	853 words/s	DECst. 3100	12.98 w/s	66
			SPARC10	35.02 w/s	24
template-based ray-casting algorithm	RIPP-10	0.166 s	486 DX2, 66 MHz	20 s	120
JPEG, 704x576 RGB image, 24 bits per pixel	MoM-3	64 ms	SPARC10/51	1.51	24
3x3 2d FIR filter, 1280x1024 8-bit	MoM-3	5.8 ms	SPARC10/51	0.71 s	122
5x5 2d FIR filter, 1280x1024 8-bit	MoM-3	9.6 ms	SPARC10/51	1.87 s	180
7x7 2d FIR filter, 1280x1024 8-bit	MoM-3	15 ms	SPARC10/51	3.58 s	213
Ising 128 lattice, 1000 iterations	MoM-3	3.85 s	SPARC10/51	1208 s	331
Calorimeter	Enable++ (estimation)	10 µs execution time	modern RISC (PowerPC, Alpha, Sparc10		430
Radiation Tracker TRT	Enable++ (estimation)	9 µs execution time			180
Silicon Tracker SCT	Enable++ (estimation)	4 µs execution time			600
sequence comparison algorithm	Splash	20 ms	CM-2	4.7 s	235
			Cray 2	6.5 s	325
			Sun 3/140	48 s	2400
convolution 3 x 3, on 512 x 512 pixel image	Splash2	13 ms, coeff. are powers of 2 (65 ms, otherwise)	SPARC20/40	1.7 s	131 (26)
			SPARC10/30	3.0 s	230 (46)
			SPARC2	4.5 s	346 (69)
			i-860	51.9 ms	4 (0.8)
DNA sequence comparison	Splash2 (16 modules)	43000 M CUPS (cells updated per second)	MP-1	32 M	1344
			CM-2	5.9 M	7288
			Sun 10/30GX	1.2 M	35833



massively parallel universal multi computers or multi processors are an illusion
communication process (organized at run time: highly overhead-prone)

parallel data paths:
communication structure (organized at compile time: no run time overhead)

von Neumann: control-driven
systolic (no control): driven by pre-organized data streams (no branching)
„data flow machines": driven by arbitration (indeterministic)
xputers: driven by data sequencing (deterministic)

instruction set defined:

µP: duality instruction sequencer/ ALU (compact instruction code = bottleneck - multiplexer = bottleneck)

ASIP: by synthesis tools

CCM: by duality instruction sequencer/ ALU and (extensions) by synthesis tools

PACCM:

ASAP: by synthesis tools

communication defined:

µP: as a process at run time (PaR)

ASIP: as a process at run time (PaR)

CCM: as a process at run time (PaR)

PACCM: as a structure at compile time (SaC)

ASAP: as a structure at synthesis time (SaS)