Soft-Core CPUs

An inventory of ~600 designs

James C. Brakefield

April 17, 2018

What is a soft-core CPU?

- Written in VHDL, Verilog or other Register Transfer Language (RTL)
- Can be compiled, placed, routed and downloaded into a FPGA
- Provides a digital processor that runs a stored program, has data memory and IO
- Data width, instruction set and memory space(s) all per the design

From LEM1_9ptr.vhd:

```
case opcode is
    when op_LD  (8 downto 6) => accn<=        bitRAMw_DO;          accwe<='1';
    when op_LDC (8 downto 6) => accn<=    NOT bitRAMw_DO;          accwe<='1';
    when op_AND (8 downto 6) => accn<=acc AND bitRAMw_DO;          accwe<='1';
    when op_OR  (8 downto 6) => accn<=acc OR  bitRAMw_DO;          accwe<='1';
    when op_XOR (8 downto 6) => accn<=acc XOR bitRAMw_DO;          accwe<='1';
    when op_ADC (8 downto 6) => accn<=acc XOR bitRAMw_DO XOR cry; accwe<='1';
    cryn<=(acc AND cry)OR(acc AND bitRAMw_DO)OR(cry AND bitRAMw_DO); crywe<='1';
    when op_INCM(8 downto 6) => bitRAMw_DI<=bitRAMw_DO XOR cry;   bitram_we<='1';
    when op_MACS(8 downto 6) =>
```

FPGA chip resources

Sea of **LookUp Tables** (can generate any Boolean function of up to six inputs)

Flip-flops, one or two per LUT

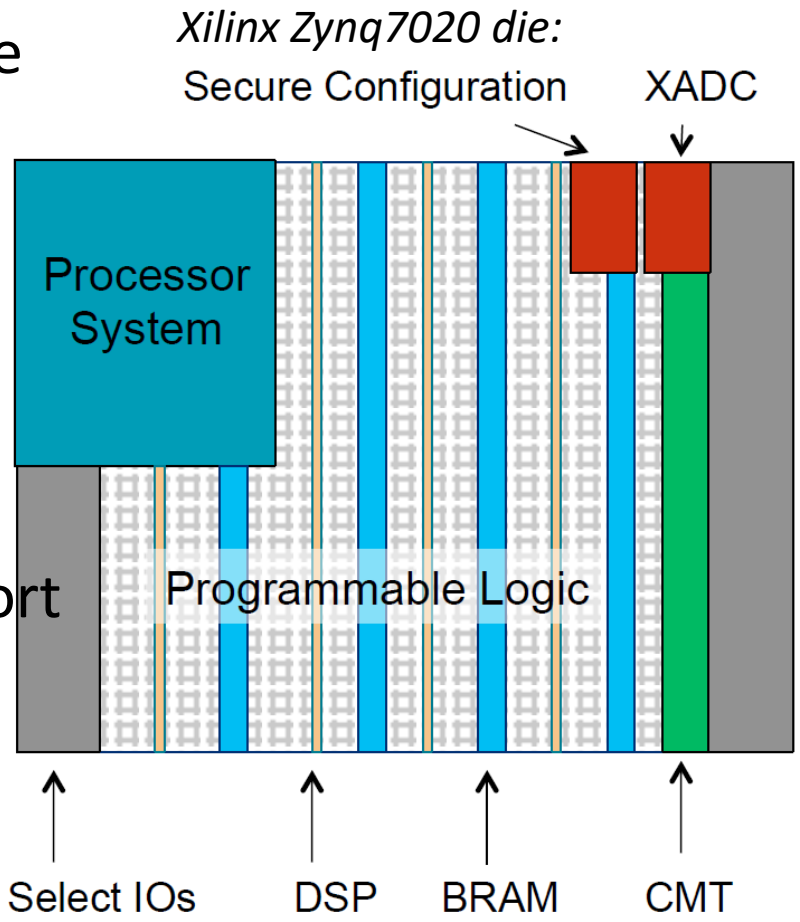
IO pins, various voltage standards

Adders, variable length, use LUTs

LUT and Block RAMs, often dual port

Fixed size **multipliers**: 18 by 18 and larger, now include 48-64 bit adder/ALU/DSP

Clock generators (CMT)



Soft uP Economics

- Current low cost FPGAs provide 1K LUTs per dollar in quantity one
  - Suppliers: Digikey, Mouser, Avnet, …
- Many embedded processor applications have modest memory requirements
- A capable uP can fit into 300 LUTs: that's 30 cents and if the FPGA is already part of the project, the additional cost can be zero

Soft-Core sources

Web search on:

- “VHDL CPU”
- “Verilog CPU”
- “FPGA soft core processor design”
- “github *author’s name*”

www.opencores.org

www.librecores.org/project/list

Inventory list spreadsheets (see recent PDFs)

- opencores.org/project/up_core_list/downloads

What's out there?

- Student projects and educational designs
  - Usually a small ISA, documentation varies
- Hobby projects & loss leaders ☺
  - Way to build skills/reputation & get attention
  - A popular design can form a business
- Proprietary designs
  - Commercial IP: both for ASICs and FPGAs
  - Published papers, books, teaching materials
- Unfinished, stalled, low quality ☹

Soft uP spreadsheet

Spreadsheet upper left corner, sorted by design name

C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
_uP_all_soft folder	opencores or prmary link	status	author	style / clone	data size	inst siz	FPGA	repo rter	com men	LUTs ALUT	LUT?	mult s	blk ra	F max	date	tool ver	MIPS /inst	clks /	KIPS /LUT
6809_6309	https://opencc	beta	Alejandro Paz Schmie	6809	8	8	stratix-5	James Brake		1711	A			223	##	q14.0	0.33	3.0	14.3
68hc05	https://opencc	stable	Ulrich Riedel	6805	8	8	kintex-7-3	James Brake		1225	6			300	##	14.7	0.33	4.0	20.2
68hc08	https://opencc	stable	Ulrich Riedel	6808	8	8	kintex-7-3	James Brake		2290	6			101	##	14.7	0.33	4.0	3.6
8bit_chapmar	http://www.ec	beta	Rob Chapman, Steve	forth	8	8	kintex-7-3	James Brake		176	6			131	##	14.7	0.33	1.0	245.5
8bit_piped_pi	https://opencc	stable	Mahesh Sukhdeo Pal	RISC	8	16	kintex-7-3	James	swap	1049	6		1	370	##	14.7	0.33	1.0	116.4
8bit-verilog_mcu		stable	Josh Friend	accum	8	8	kintex-7-3	James	insert	240	6		1	328	##	14.7	0.33	2.0	225.5

Spreadsheet lower right corner, sorted by design name

W	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR
ven dor	SOC	src code	# src files	top file	doc	tool chai	fltg pt	Hav'	max data	max inst	byte adrs	# inst	adr mod	# reg	pip e	start year	last revi	secondary web link	note worthy	comments
X	Y	verilog	55	top_de1	Y	yes	N	N	64K	64K	Y					####	####		Microprocessor targeting embedde	interfaces to DRAM, based
IX	Y	vhdl	19	top_s3e	Y	yes	N	N	64K	64K	Y					####	####		based on Daniel Wallner's T80	
X		verilog	37	zap_top	Y	yes	N	N	4G	4G	Y			16		####	####	ddi0100e_armv1-5	ARMv4T & Thumbv1	has cache & mmu
X		verilog	32	fpga_zet	Y	yes	N	N	1M	1M	Y					####	####	http://zet.aluzina.o	equivalent to 80186, boots MS-DO	Zet The x86 (IA-32) open im
X		verilog	7	zipcpu	Y		N	N	4G	4G	N	20		16	5	####	####	www.librecores.org	ISA has chnaged, multiple instruction sequences for many oper	
		system	15	plugh	Y		N									####		http://inform-fictio	Z-machine (Zork)	https://www.youtube.com/
X		vhdl	23	zpu_core	Y	yes	N		4G	4G	Y	37				####	####		zpu4: 16 & 32 bit versions, code si	ZPU the worlds smallest 32
X	Y	vhdl		papilio_p	Y	yes	N		4G	4G	Y	37				####	####		SoC version of modified ZPU	pipelined, removed ucf file
I	Y	vhdl	53	ztachip												####	####		multi-core with MIPS master	files no longer available, wa

CPU Categories

- Legacy processors, AKA clones *# of designs*
  - 50% of inventory 300
- Originals 189
  - RISC 99
  - Stack/Forth 51
  - Accumulator 30
  - Other 9
- Lacking open source 142
  - Planning, paper only, proprietary, other

RTL Languages

• VHDL	215
• Verilog	193
• System Verilog	17
• Spinal/Scala	7
• VHDL & Verilog	6
• MyHDL (Python)	3
• Schematic	2
• Proprietary (don’t know)	27

Main Stream soft uPs

- NIOS2 Altera-Intel 32-bit RISC
- microBlaze Xilinx 32-bit RISC
- MIPS all variety of subsets 32-bit RISC
- RISC-V Un. Calif. Berkeley 32-bit RISC
- https://en.wikipedia.org/wiki/Soft_microprocessor
- Legacy uPs: 6502, 8051, AVR, x86, z80, …

Most Clones

MIPS	25	AVR	10	ARM7	5
RISC-V	22	Z80	10	8080	5
6502	19	68000	9	6809	4
PIC16	14	uBlaze	9	PDP-11	4
openRISC	12	6800	7	PDP-8	3
X86	11	picoBlaze	7	MSP430	3
8051	11	SPARC	7	Others	22

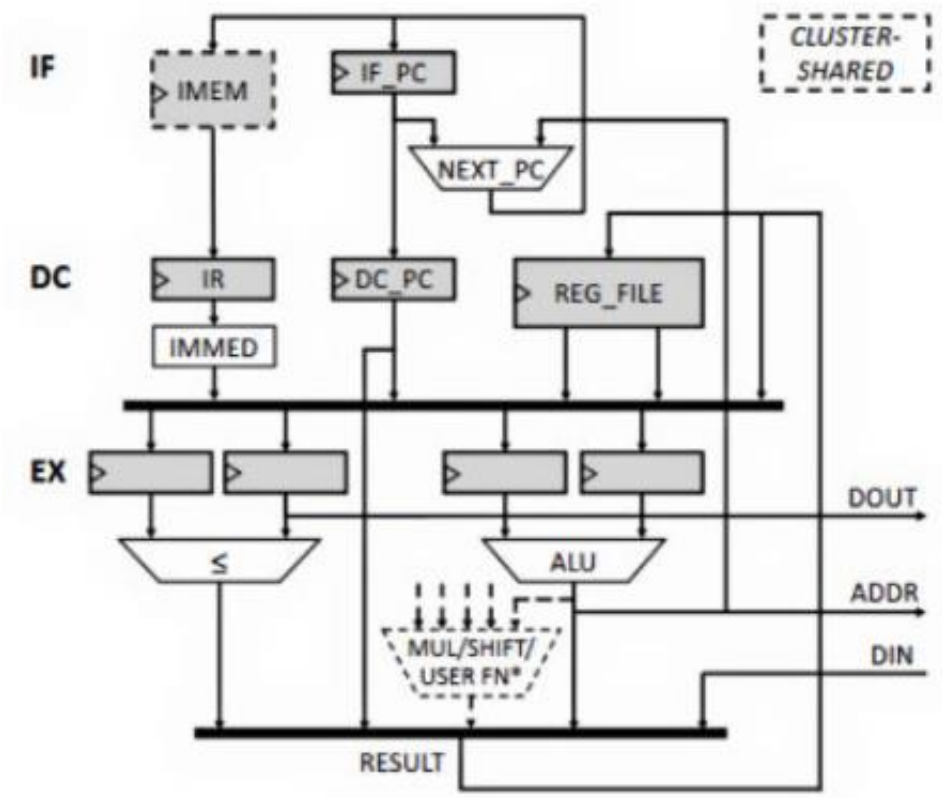
Figure of Merit caveats

- Work done per resources used
  - Proportional to clock speed
  - ~Proportional to data width:
    - 32-bits: 1.0, 24-bits: 0.83, 16-bits: 0.67, 8-bits: 0.33
  - Inversely proportional to clocks/instruction
  - Inversely proportional to LUTs used
- Gotchas:
  - No adjustment for bare-core memory access time
  - Rough adjustment for ISA capability
  - No adjustment for memory size
  - No adjustment for floating-point
- FoM decreases as instructions & peripherals added

32-bit High Figure of Merit Designs

Name	comment	LUTs	Fmax	FOM
GRVI-phalanx	uses DSP as ALU	320	375	1172
Jan Gray: RISC-V ISA, 1680 cores on a large FPGA				
Microblaze	many config vars	546	320	603
Xilinx IP, best FOM, no caches, no MMU, no fltg-pt				
NIOS2	many config vars	895	310	390
Altera IP, best FOM, no caches, no MMU, no fltg-pt				
Hard core **ARM**	using chip area	4500	1050	583
Both Altera and Xilinx, area includes caches & fltg-point				

GRVI-phalanx



A GRVI Processing Element

https://forums.xilinx.com/t5/Xcell-Daily-Blog/Shazam-Jan-Gray-gets-1680-RISC-V-processors-to-dance-on-the-head/ba-p/789549

"Typical of a Jan Gray design, the GRVI processing element is hand-mapped and – floorplanned into the UltraScale+ architecture and then stamped 1680 times"

16-bit High FoM Designs

Names	style	LUTs	Fmax	FOM
Leros	accum	112	182	1089
Lutiac	register	140	198	948
iDEA	RISC	321	405	845
Octavo	register	500	550	737
P16b	forth	367	355	648
Xr16	RISC	273	263	645
J1a	forth	518	412	636

8-bit High FoM Designs

Names	style	LUTs	Fmax	FOM
ssbcc	forth	196	474	798
avr8	AVR	174	418	792
Myrisc1	RISC	121	231	629
lwrisc	accum	88	230	444
popcorn	accum	267	347	428

Most Prolific Authors

Robert Finch	12	Aleksander Osman	3
John Kent	6	Han Tiggeler	3
Daniel Wallner	5	Jose Ruiz	3
Shawn Tan	4	Lazaridis Dimitris	3
Ulrich Riedel	4	James Brakefield	3
C.H. Ting	4		
Stephen Nolting	3		

RTL Styles

Two process	LEON3	Jiri Gaisler[1]
Handcrafted	XR16	Jan Gray
Plan data path, add ctrl logic		Jan Gray
uCode	mcl65	Ted Fried
Single stage pipe	mips	Harris & Harris[2]
Op-code case statement	LEM1_9	Jim Brakefield
Everything in a single file	Table887	Robert Finch

[1]www.gaisler.com/doc/structdesign.pdf: A Structured VHDL Design Method
[2]Harris & Harris 2012: Digital Design and Computer Architecture 2nd ed.

Significant Designs

- Jan Gray XR16, GRVI-phalanx
  - High density & high performance
- Jiri Gaisler LEON2-3 (SPARC7)
  - Separate combinatorial & register update process
- Ted Fried mcl51, mcl65, mcl86
  - Uses micro-code to dramatically reduce LUT count
- Hirotsugu Nakano mincpu, tiny_cpu
  - Use of Flex, Bison & Perl to generate gcc compiler

Robert Finch: Table887.v

```
// Table887.v
//         __
//   \\__/ o\    (C) 2014  Robert Finch, Stratford
//    \  __ /    All rights reserved.
//     \/_//     robfinch<remove>@finitron.ca
//       ||
// This source file is free software: you can redistribute it
and/or modify  it under the terms of the GNU Lesser General
Public License
...
// Register-Register Format:
//    5     3   3    3     2
// |  2  | Ra | Rb | Rt | Fn |
// +-----+----+----+----+----+
// conditional branch format:
//    5     3     8
// | 16  | Ra |  disp  |
// +-----+----+--------+
```

Robert Finch: Table887.v cont'd

```
`define RR1             5'd1
`define RR2             5'd2
`define ADD             2'd0
`define SUB             2'd1
`define ADDI    5'd4
`define SUBI    5'd5
...
// IFETCH Stage
IFETCH1:
        begin
                wb_read(pc);
                pc <= pc + 24'd2;
                next_state(IFETCH2);
        end
...
```

Robert Finch: Table887.v cont'd

```
// EXECUTE Stage
EXECUTE:
        begin
                next_state(MEMORY);
                case(ir[7:0])
                // Arithmetic / Logical
                `RR2:
                case(ir[15:14])
                        `ADD:   res <= a + b;
                        `SUB:   res <= a - b;
                        `CMP:   res <= as < bs ? ...
                        endcase
                `RR3:
                case(ir[15:14])
                        `AND:   res <= a & b;
                        `OR:    res <= a | b; ...
                        endcase
                `ADDI:  res <= a + immediate;
                `SUBI:  res <= a - immediate; ...
```

Jim Brakefield

- LEMx_9ptr (Logic Emulation Machine)
  - For slow logic, logic emulation uses least resources
  - 1-bit and 4-bit data widths
  - 9 & 18-bit instructions, four memory pointers

    `xxxr-sspp` instructions using pointers: (p),(p++),(--p)

    `xxxr-11pp nnnnnnnnn` instructions with pointer offset
- ROIS24_24 (Register Oriented Instruction Set)
  - 24-bit instruction: as capable as 32-bit & gives competitive code density without needing to introduce a 16-bit subset
  - 8, 16 and 24/32-bit data

`xxxxxx uddddd zrrrrr –sssss`

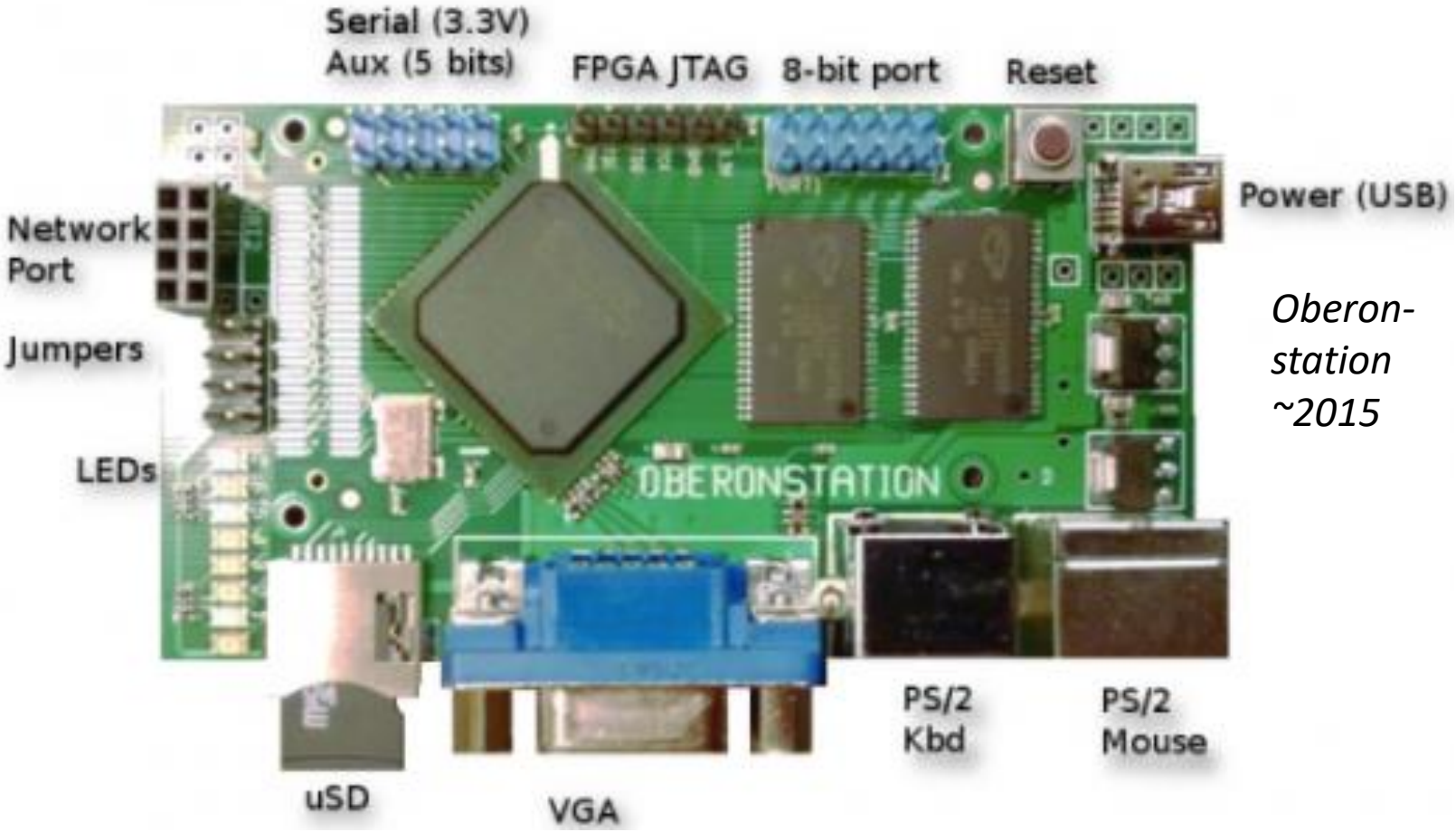
`xxxxxx uddddd snnnnn nnnnnn`

`xxxxxx uddddd zrrrrr nnnnnn`

`xxxxxx nnnnnn nnnnnn nnnnnn`

Niklaus Wirth

- Created Pascal & several other languages
- A series of workstations using Oberon language and Oberon OS
- History of very compact OS, utilities & compiler
- His Risc5 soft uP is smallest design that includes floating-point



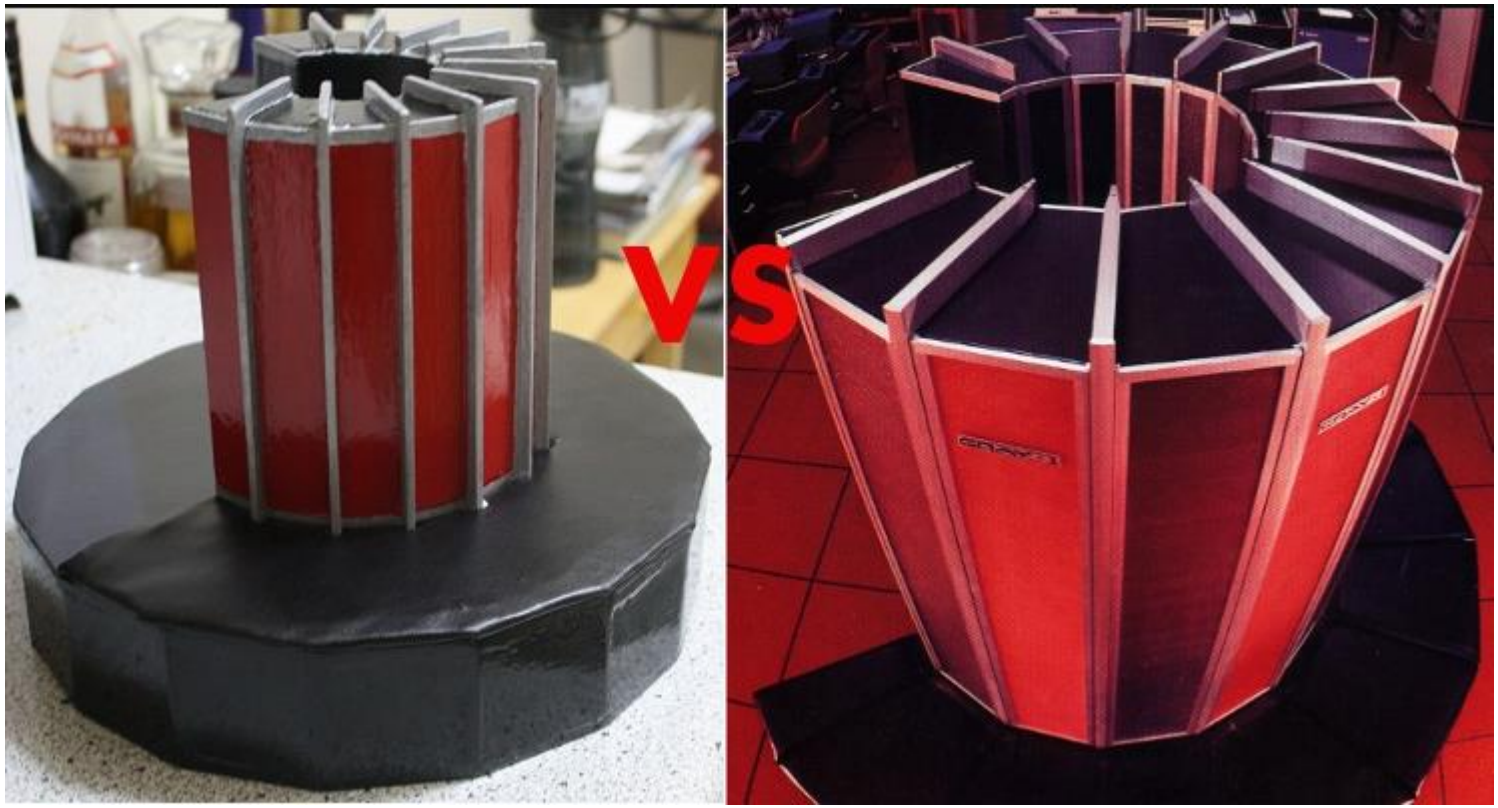
Oberon-station ~2015

Christopher Fenton

Soft CRAY-1 on a FPGA card

Built CRAY-1 lookalike cabinet for the FPGA card

http://www.chrisfenton.com/homebrew-cray-1a/



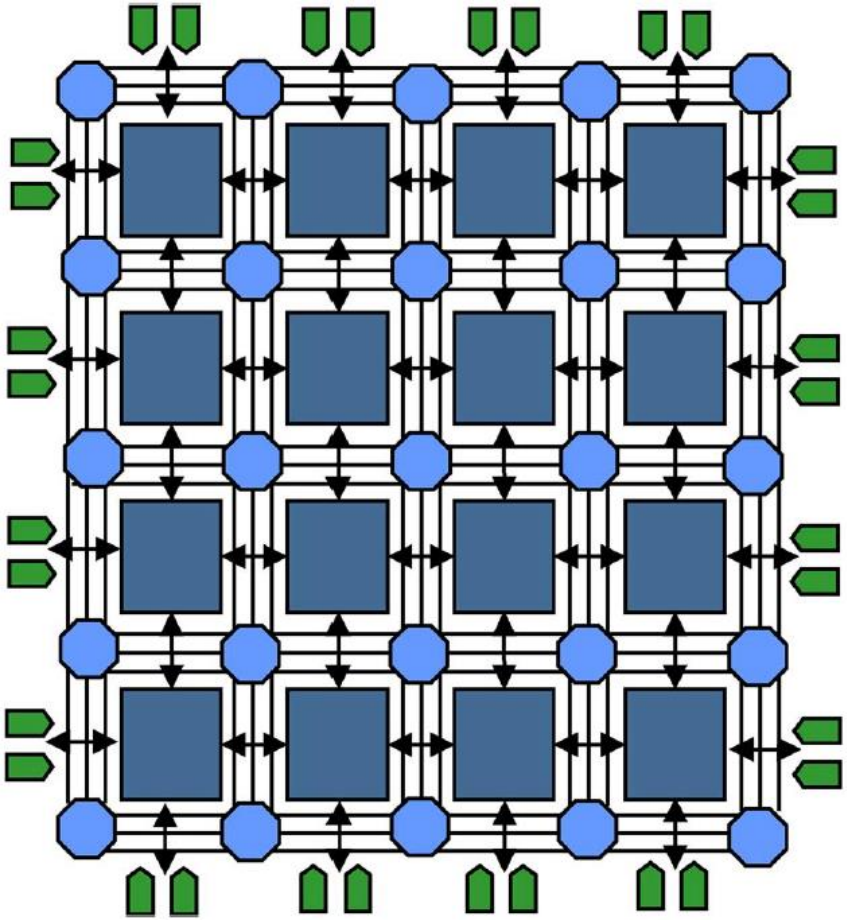
Single clock design

- Read the next instruction from block RAM
- Decode the instruction
- Read out the operands from the LUT RAM register file
- Run operands thru the ALU
- Form the next instruction address
- Clock in the instruction address and ALU result
- Repeat

A two stage pipeline design

- Stage 1: instruction read & decode
  - Read the next instruction from block RAM
  - Decode the instruction
  - Increment PC
  - Clock in decoded instruction and new PC
- Stage 2: execution of previous instruction
  - Read out the operands from the LUT RAM register file
  - Run operands thru the ALU
  - Clock in the ALU result

Generic FPGA



Generic FPGA diagram

Green: IOs

Squares: LUT groups/slices/blocks

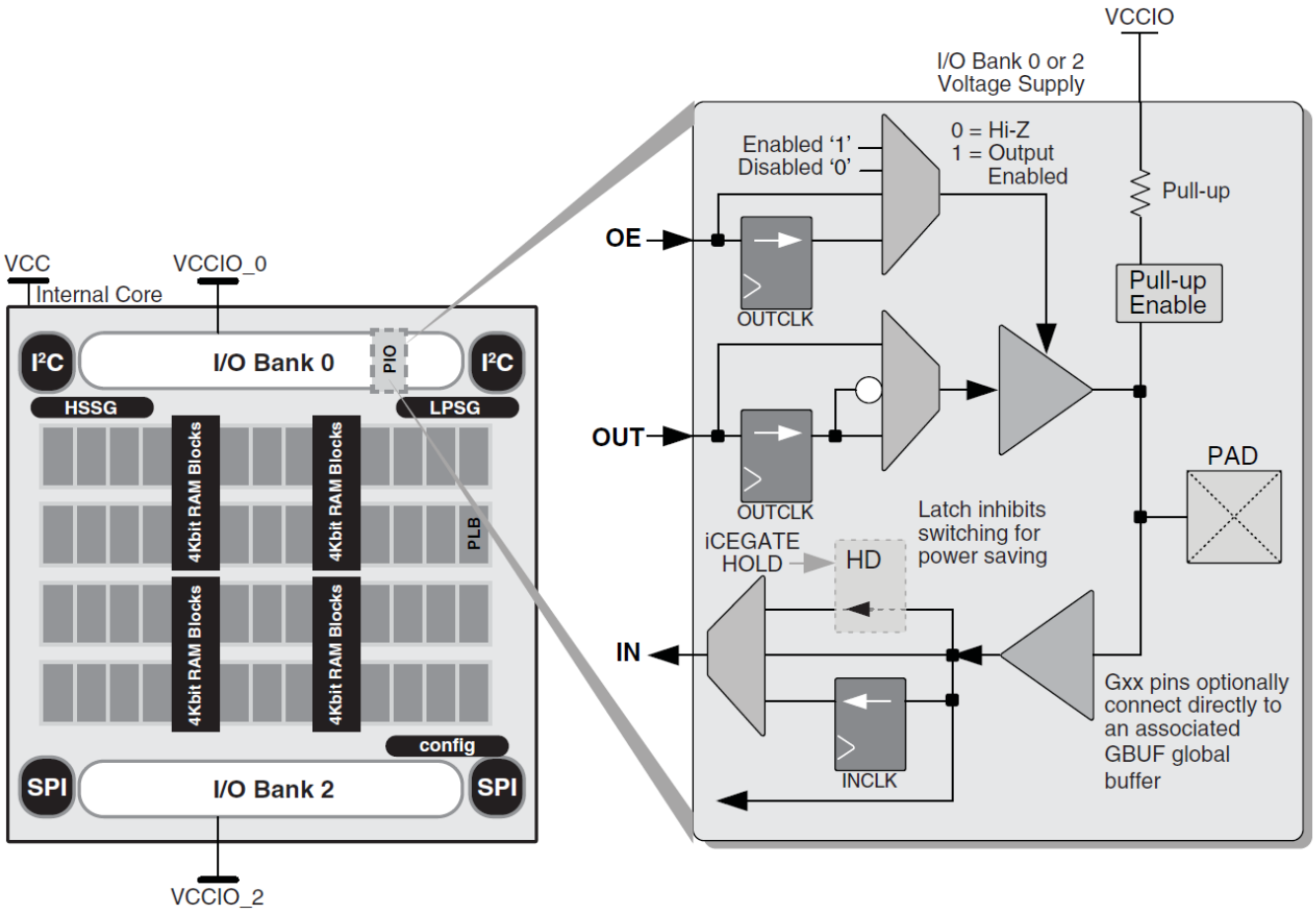
Octagons: Wiring interconnections

Lines: Wire segment bundles

From: Trimberger, 2015 Proc IEEE v103 #03 pg320

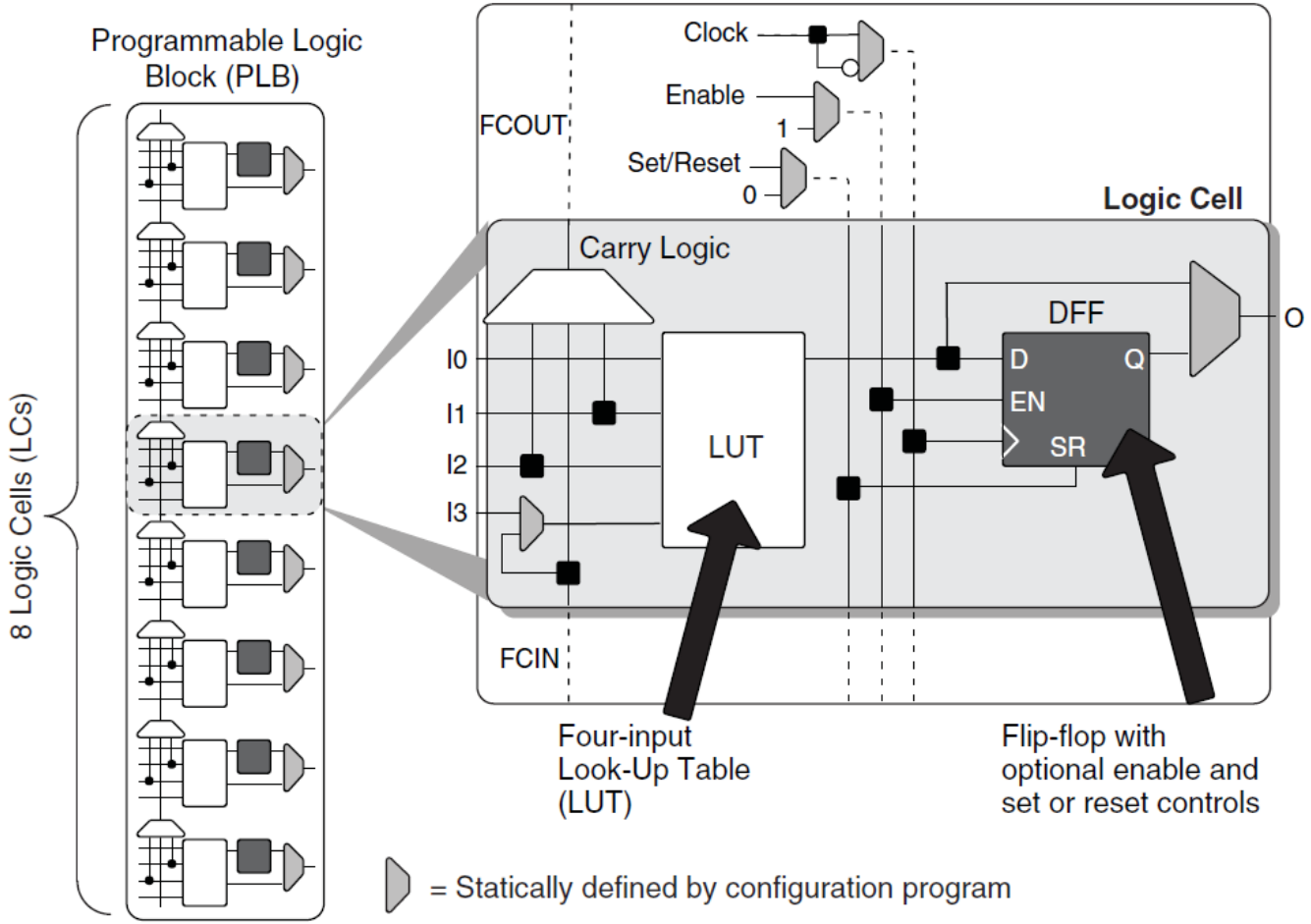
“ASIC” IO

Global input buffers used for clocks and reset



Lattice Semiconductor iCE40FamilyHandbook.pdf pg6-7

LUT + DFF



3LUTs tried
6LUTs in use
ALM: ~two 5LUTs with multiple configurations

4LUT: 16 bits of memory & a 16:1 MUX

Lattice Semiconductor iCE40FamilyHandbook.pdf pg6-2

Block RAM

- Variable aspect ratio (on each port)

  16Kx1, 8Kx2, 4Kx4, 2Kx8/9, 1Kx16/18, 512x32/36

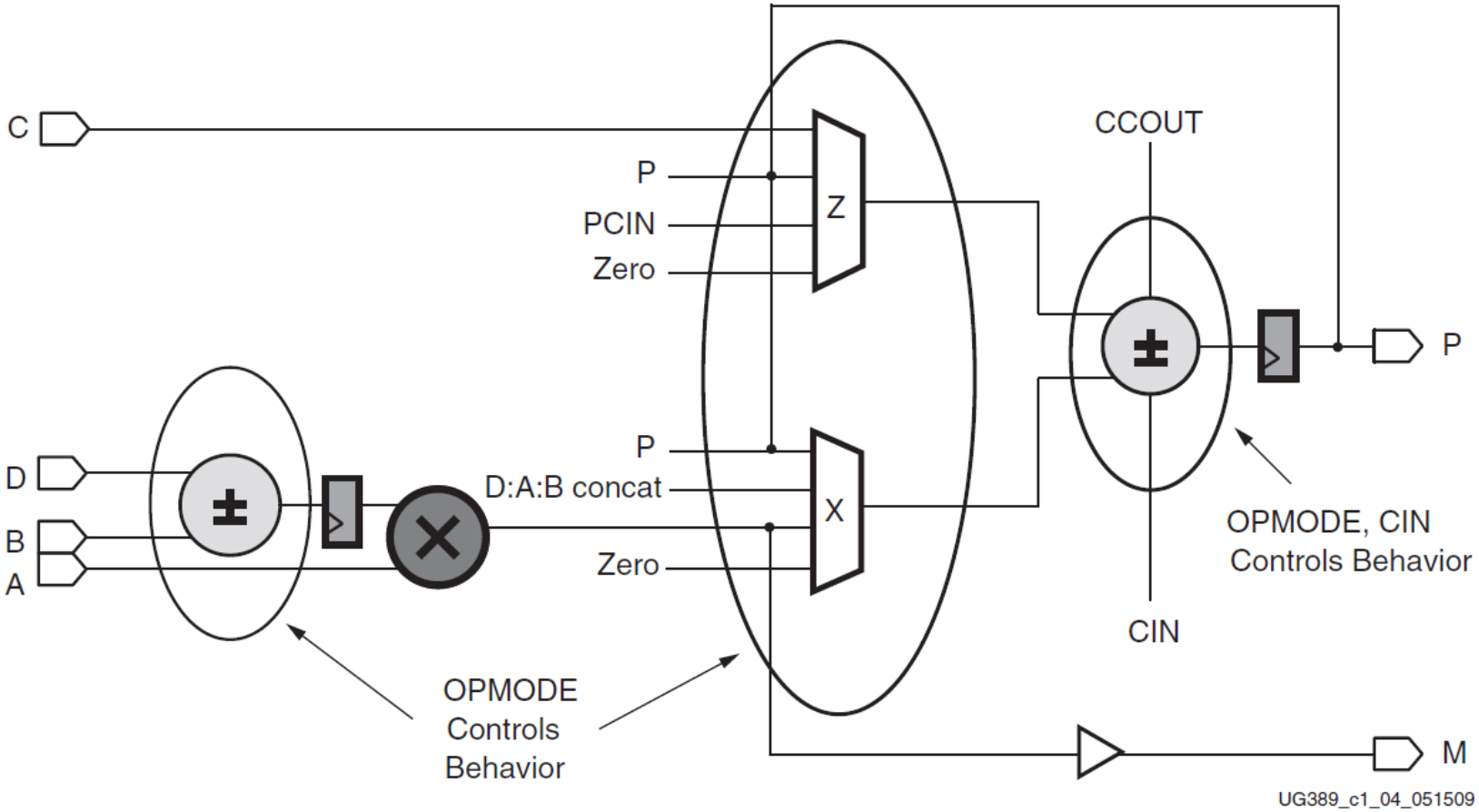
- A variety of RAM capacities

  LUT RAM (16x1), small block RAM (~32x18), block RAM (~512x36), large block RAM (~4Kx72)

- Uses:

  Buffers, FIFOs, shift registers, scratch pad memory, DSP coefficients, two single port RAMs, u-code

Simplified DSP slice



Xilinx Spartan-6 FPGA DSP48A1 Slice pg 17