

CS184a: Computer Architecture (Structure and Organization)

Day 22: March 4, 2005
Control



Previously

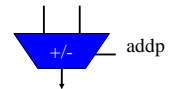
- Looked broadly at instruction effects
- Looked at structural components of computation
 - interconnect
 - compute
 - retiming
- Looked at time-multiplexing

Today

- Control
 - data-dependent operations
- Different forms
 - local
 - instruction selection
- Architectural Issues

Control

- **Control:** That point where the data affects the instruction stream (operation selection)
 - Typical manifestation
 - data dependent branching
 - if (a!=0) OpA else OpB
 - bne
 - data dependent state transitions
 - new => goto S0
 - else => stay
 - data dependent operation selection

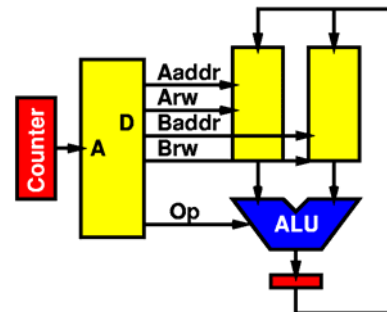


Control

- **Viewpoint:** can have instruction stream sequence without control
 - *l.e.* static/data-independent progression through sequence of instructions is control free
 - C0→C1→C2→C0→C1→C2→C0→...
 - Similarly, FSM w/ no data inputs
 - *E.g.* Day 5...non-branching datapath

Day 5

Our “First” Programmable Architecture



Terminology (reminder)

- **Primitive Instruction (*pinst*)**
 - Collection of bits which tell a bit-processing element what to do
 - Includes:
 - select compute operation
 - input sources in space (interconnect)
 - input sources in time (retiming)
- **Configuration Context**
 - Collection of all bits (*pinsts*) which describe machine's behavior on one cycle

Caltech CS184 Winter2005 -- DeHon

7

Why?

- Why do we need / want control?
- Static interconnect sufficient?
- Static sequencing?
- Static datapath operations?

Caltech CS184 Winter2005 -- DeHon

8

Back to "Any" Computation

- Design must handle all potential inputs (computing scenarios)
- Requires sufficient generality
- However, *computation for any given input may be much smaller than general case.*
- **Instantaneous** computation \ll potential computation

Caltech CS184 Winter2005 -- DeHon

9

Screwdriver Analogy

- Need capability to handle
 - Slothead
 - Phillips
 - Torq
 - Hex...
- But only need one at a time...

Caltech CS184 Winter2005 -- DeHon

10

Video Decoder

- *E.g.* Video decoder [frame rate = 33ms]
 - if (packet==FRAME)
 - if (type==I-FRAME)
 - I-FRAME computation
 - else if (type==B-FRAME)
 - B-FRAME computation

Caltech CS184 Winter2005 -- DeHon

11

Packet Processing

- If IP-V6 packet
 - ...
- If IP-V4 packet
 - ...
- If VoiP packet
 - ...
- If modem packet
 - ...

Caltech CS184 Winter2005 -- DeHon

12

Two Control Options

- Local control
 - unify choices
 - build all options into spatial compute structure and select operation
- Instruction selection
 - provide a different instruction (instruction sequence) for each option
 - selection occurs when choose which instruction(s) to issue

Caltech CS184 Winter2005 -- DeHon

13

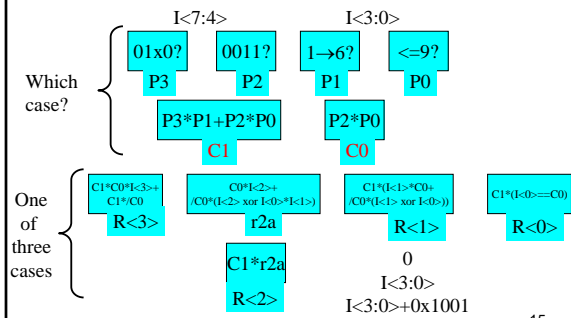
Example: ASCII Hex→Binary

- If $(c \geq 0x30 \ \&\& \ c \leq 0x39)$
 - $res = c - 0x30$ // $0x30 = '0'$
- elseif $(c \geq 0x41 \ \&\& \ c \leq 0x46)$
 - $res = c - 0x41 + 10$ // $0x41 = 'A'$
- elseif $(c \geq 0x61 \ \&\& \ c \leq 0x66)$
 - $res = c - 0x61 + 10$ // $0x61 = 'a'$
- else
 - $res = 0$

Caltech CS184 Winter2005 -- DeHon

14

Local Control



Caltech CS184 Winter2005 -- DeHon

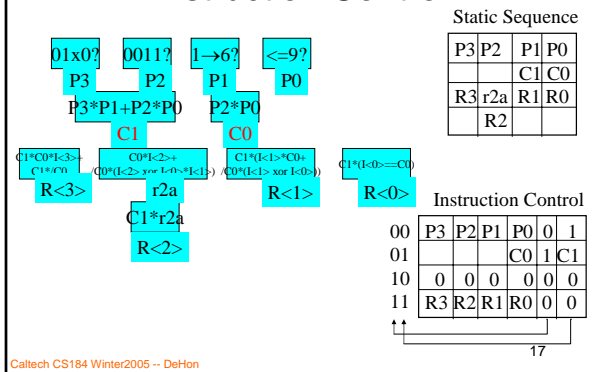
Local Control

- LUTs used \neq LUT evaluations produced
- This example:
 - each case only requires 4 4-LUTs to produce $R<3:0>$
 - takes 5 4-LUTs once unified
- \Rightarrow Counting LUTs not tell cycle-by-cycle LUT needs

Caltech CS184 Winter2005 -- DeHon

16

Instruction Control



Caltech CS184 Winter2005 -- DeHon

Static/Control

- Static sequence
 - 4 4-LUTs
 - depth 4
 - 4 context
 - maybe 3
 - shuffle r2a w/ C1, C0
 - execute 0 1 1 2 0 1 1 2
- Control
 - 6 4-LUTs
 - depth 3
 - 4 contexts
 - Example too simple to show big savings...

Caltech CS184 Winter2005 -- DeHon

18

Local vs. Instruction

- If can decide early enough
 - and afford schedule/reload
 - instruction select → less computation
- If load too expensive
 - local instruction
 - faster
 - maybe even less capacity (AT)

Caltech CS184 Winter2005 -- DeHon

19

Slow Context Switch

- Instruction selection profitable only at coarse grain
 - Xilinx ms reconfiguration times
 - HSRA μ s reconfiguration times
 - still 1000s of cycles
- E.g. Video decoder [frame rate = 33ms]
 - if (packet==FRAME)
 - if (type==I-FRAME)
 - IF-context
 - else if (type==B-FRAME)
 - BF-context

Caltech CS184 Winter2005 -- DeHon

20

Local vs. Instruction

- For multicontext device
 - i.e. fast (single) cycle switch
 - factor according to available contexts
- For conventional devices
 - factor only for gross differences
 - and early binding time

Caltech CS184 Winter2005 -- DeHon

21

Optimization

- Basic Components
 - T_{load} -- config. time
 - T_{select} -- case compute time
 - T_{gen} -- generalized compute time
 - A_{select} -- case compute area
 - A_{gen} -- generalized compute area
- Minimize Capacity Consumed:
 - $AT_{local} = A_{gen} \times T_{gen}$
 - $AT_{select} =$
 - $A_{select} \times (T_{select} + T_{load})$
 - $T_{load} \rightarrow 0$ if can overlap w/ previous operation
 - know early enough
 - background load
 - have sufficient bandwidth to load

Caltech CS184 Winter2005 -- DeHon

22

FSM Control Factoring Experiment

Caltech CS184 Winter2005 -- DeHon

23

FSM Example

- FSM -- canonical "control" structure
 - captures many of these properties
 - can implement with deep multicontext
 - instruction selection
 - can implement as multilevel logic
 - unify, use local control
- Serve to build intuition

Caltech CS184 Winter2005 -- DeHon

24

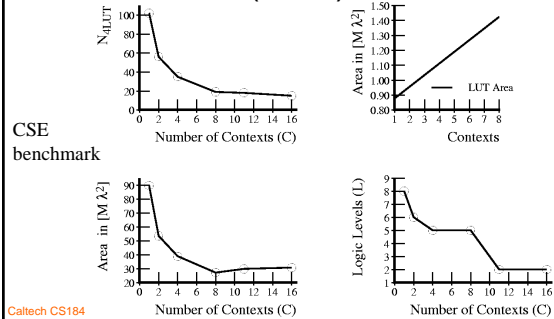
Full Partitioning

- Full partitioning comes out better
 - ~40% less area
- Note: full partition may not be optimal area case
 - e.g. intro example,
 - no reduction in area or time beyond 2-context implementation
 - 4-context (full partition) just more area
 - (additional contexts)

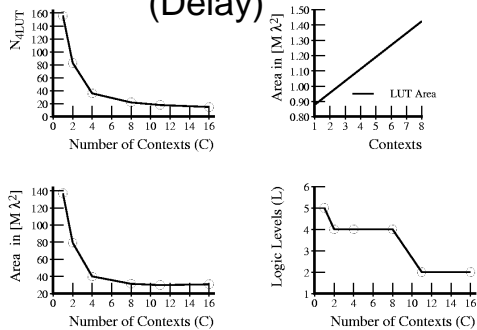
Caltech CS184 Winter2005 -- DeHon

31

Partitioning versus Contexts (Area)



Partitioning versus Contexts (Delay)



Caltech CS184

Partitioning versus Contexts (Heuristic)

- Start with dense mustang state encodings
- Greedily pick state bit which produces
 - least greatest area split
 - least greatest delay split
- Repeat until have desired number of contexts

Caltech CS184 Winter2005 -- DeHon

34

Partition to Fixed Number of Contexts

FSM	States	Best Single Context	Area Ratio by Number of Context Dense Encodings							
			1	2	4	8	16	32	64	
Area Target										
average ratio		1.00	1.51	0.86	0.63	0.56	0.70	1.09	1.92	
average delta		0.00	-0.27	0.33	1.27	2.18	2.70	3.03	3.06	
Delay Target										
average ratio		1.00	1.45	1.05	0.59	0.50	0.62	0.95	1.67	
average delta		0.00	-0.91	-0.48	0.06	0.64	0.91	1.15	1.21	

N.B. - more realistic, device has fixed number of contexts.

Caltech CS184 Winter2005 -- DeHon

35

Extend Comparison to Memory

- Fully local => compute with LUTs
- Fully partitioned => lookup logic (context) in memory and compute logic
- How compare to fully memory?
 - Simply lookup result in table?

Caltech CS184 Winter2005 -- DeHon

36

Memory FSM Compare (small)

FSM	states	ins	outs	Min area (Mλ ²)	Integral Addr. & Data Organization	Memory area (Mλ ²)	FPGA area (Mλ ²)	8-ctx DPGA area (Mλ ²)
bbtas	6	2	2	0.1	2 ⁵ ×5	0.2	6.1	7.1
dk15	4	3	5	0.3	2 ⁵ ×7	0.3	21.9	10.0
dk17	8	2	3	0.2	2 ⁵ ×6	0.2	16.7	8.5
dk512	15	1	3	0.3	2 ⁵ ×7	0.3	17.6	10.0
mc	4	3	5	0.3	2 ⁵ ×7	0.3	7.0	10.0
modulo12	12	1	1	0.1	2 ⁵ ×5	0.2	10.5	7.1
beecount	7	3	4	0.5	2 ⁵ ×7	0.5	12.3	10.0
dk14	7	3	5	0.5	2 ⁵ ×8	0.6	50.9	11.4
dk16	27	2	3	1.0	2 ⁷ ×8	1.3	70.2	11.4
dorfile	24	2	1	0.7	2 ⁷ ×6	0.9	40.4	8.5
s27	6	4	1	0.5	2 ⁷ ×4	0.6	4.4	5.7
s8	5	4	1	0.4	2 ⁷ ×4	0.6	10.5	5.7
bbara	10	4	2	1.2	2 ⁸ ×6	1.8	21.9	11.4
ex6	8	5	8	3.4	2 ⁸ ×11	3.4	50.0	15.7
ex4	14	6	9	14.0	2 ¹⁰ ×13	16.0	18.4	18.5
bbse	16	7	7	27.0	2 ¹¹ ×11	27.0	43.9	21.4
cse	16	7	7	27.0	2 ¹¹ ×11	27.0	72.9	27.1
tbk	32	6	3	19.7	2 ¹¹ ×8	19.7	298.5	68.4

Memory FSM Compare (large)

FSM	states	ins	outs	Min area (Mλ ²)	Integral Addr. & Data Organization	Memory area (Mλ ²)	FPGA area (Mλ ²)	8-ctx DPGA area (Mλ ²)
sse	16	7	7	27.0	2 ¹¹ ×11	27.0	43.9	21.4
s386	13	7	7	22.9	2 ¹¹ ×11	27.0	36.9	18.5
keyb	19	7	2	20.4	2 ¹² ×7	34.4	98.3	31.3
planet	48	7	19	184.3	2 ¹⁸ ×25	245.8	131.7	54.1
pma	24	8	8	95.8	2 ¹⁸ ×13	127.8	72.0	34.2
s1	20	8	6	67.6	2 ¹⁸ ×11	108.1	120.3	62.7
s1a	20	8	6	67.6	2 ¹⁸ ×11	108.1	63.2	54.1
ex1	20	9	19	294.9	2 ¹⁴ ×24	471.9	105.4	55.5
s1488	48	8	19	368.6	2 ¹⁴ ×25	491.5	133.5	74.0
styr	30	9	10	276.5	2 ¹⁴ ×15	294.9	163.3	57.0
s208	18	11	2	309.7	2 ¹⁶ ×7	550.5	33.4	12.8
sand	32	11	9	1101.0	2 ¹⁶ ×14	1101.0	156.3	62.7
s820	25	18	19	188743.7	2 ²⁹ ×24	241591.9	80.8	64.1
s420	18	19	2	79272.3	2 ²⁴ ×7	140928.6	35.1	14.2
s510	47	19	7	384408.9	2 ²⁶ ×13	523449.1	47.4	25.6

- ### Memory FSM Compare (notes)
- Memory selected was “optimally” sized to problem
 - in practice, not get to pick memory allocation/organization for each FSM
 - no interconnect charged
 - Memory operate in single cycle
 - but cycle slowing with inputs
 - Smaller for <11 state+input bits
 - Memory size not affected by CAD quality (FPGA/DPGA is)

Control Granularity

- ### Control Granularity
- What if we want to run multiple of these FSMs on the same component?
 - Local
 - Instruction

- ### Consider
- Two network data ports
 - states: idle, first-datum, receiving, closing
 - data arrival uncorrelated between ports

Local Control Multi-FSM

- Not rely on instructions
- Each wired up independently
- Easy to have multiple FSMs
 - (units of control)

Caltech CS184 Winter2005 -- DeHon

43

Instruction Control

- If FSMs advance orthogonally
 - (really independent control)
 - context depth => product of states
 - for full partition
 - *i.e.* w/ single controller (PC)
 - must create product FSM
 - which may lead to state explosion
 - N FSMs, with S states => S^N product states
 - This example:
 - 4 states, 2 FSMs => 16 state composite FSM

Caltech CS184 Winter2005 -- DeHon

44

Architectural Questions

- How many pins/controller?
- Fixed or Configurable assignment of controllers to pins?
 - ...what level of granularity?



Caltech

Architectural Questions

- Effects of:
 - Too many controllers?
 - Too few controllers?
 - Fixed controller assignment?
 - Configurable controller assignment?

Caltech CS184 Winter2005 -- DeHon

46

Architectural Questions

- Too many:
 - wasted space on extra controllers
 - synchronization?
- Too few:
 - product state space and/or underuse logic
- Fixed:
 - underuse logic if when region too big
- Configurable:
 - cost interconnect, slower distribution

Caltech CS184 Winter2005 -- DeHon

47

Control and FPGAs

- Local/single instruction not rely on controller
- Potential strength of FPGA
- Easy to breakup capacity and deploy to orthogonal tasks
- How processor handle orthogonal tasks?
 - Efficiency?

Caltech CS184 Winter2005 -- DeHon

48

Control and FPGAs

- Data dependent selection
 - potentially fast w/ local control compared to uP
 - Can examine many bits and perform multi-way branch (output generation) in just a few LUT cycles
- μ P requires sequence of operations

Architecture Instr. Taxonomy

Control Threads (PCs)		Instructions per Control Thread		Instruction Depth		Granularity		Architecture/Examples	
0	0	n/a							Hardwired Functional Unit (e.g. ECC/EDC Unit, FP MPV)
			1						FPGA
	n	1	w						Reconfigurable ALUs
		n_c	1						Bitwise SIMD
1	c	w							Traditional Processors
		n_c	1						Vector Processors
									DPGA
	n	8	16						PADDI
		c	w						VLIW
m	n	1	1						HSRA/SCORE
	1	c	n_c	w					MSIMD
									VEGA
m	1	8	16						PADDI-2
		c	w						MIMD (traditional)

Admin

- FEEDBACK sheets
 - Course
 - EAS
- Final classes Monday/Wednesday
 - Monday: Specialization
 - Wednesday: wrapup, Q&A...

Big Ideas [MSB Ideas]

- **Control:** where data effects instructions (operation)
- Two forms:
 - local control
 - all ops resident => fast selection
 - instruction selection
 - may allow us to reduce **instantaneous** work requirements
 - introduce issues
 - depth, granularity, instruction load time

Big Ideas [MSB-1 Ideas]

- Intuition => looked at canonical FSM case
 - few context can reduce LUT requirements considerably (factor dissimilar logic)
 - similar logic more efficient in local control
 - overall, moderate contexts (e.g. 8)
 - exploits both properties
 - better than extremes
 - single context (all local control)
 - full partition
 - flat memory (except for smallest FSMs)