

CS184a: Computer Architecture (Structure and Organization)

Day 22: March 4, 2005
Control



Caltech CS184 Winter2005 -- DeHon

Previously

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

2

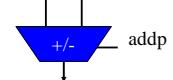
Today

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

3

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



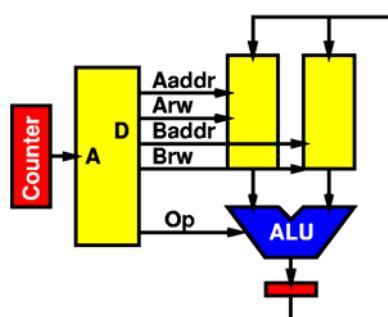
4

Control

- **Viewpoint:** can have instruction stream sequence without control
 - *i.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

5

Our “First” Programmable Architecture



6

Caltech CS184 Winter2005 -- DeHon

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?

8

Caltech CS184 Winter2005 -- DeHon

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 << potential computation

Caltech CS184 Winter2005 -- DeHon

9

Screwdriver Analogy

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

10

Caltech CS184 Winter2005 -- DeHon

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
 -

12

Caltech CS184 Winter2005 -- DeHon

Two Control Options

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

Caltech CS184 Winter2005 -- DeHon

13

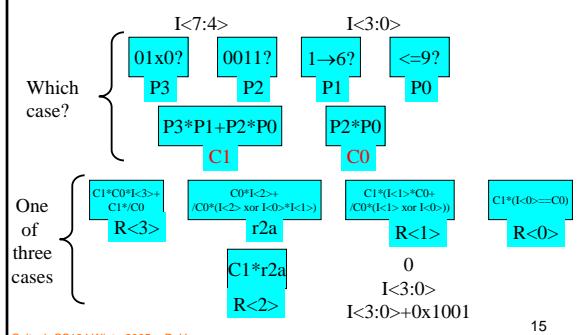
Example: ASCII Hex→Binary

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

Caltech CS184 Winter2005 -- DeHon

14

Local Control



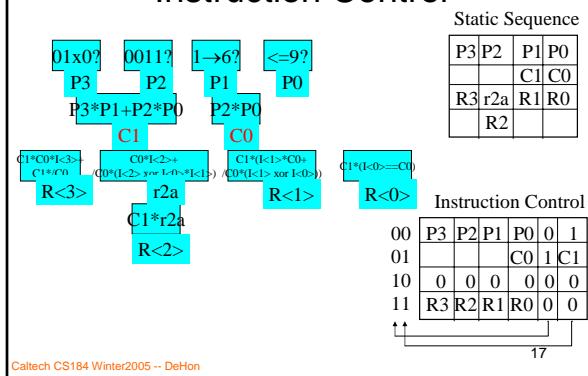
Local Control

- 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
- => Counting LUTs not tell cycle-by-cycle LUT needs

Caltech CS184 Winter2005 -- DeHon

16

Instruction Control



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)

19

Caltech CS184 Winter2005 -- DeHon

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

20

Caltech CS184 Winter2005 -- DeHon

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

21

Caltech CS184 Winter2005 -- DeHon

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

22

Caltech CS184 Winter2005 -- DeHon

FSM Control Factoring Experiment

23

Caltech CS184 Winter2005 -- DeHon

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

24

Caltech CS184 Winter2005 -- DeHon

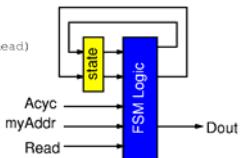
FSM Example (local control)

FSM Description

```

Idle (00):
  if (Acyc & myAddr & Read)
    goto Wait1
  else
    goto Idle
Wait1 (01):
  goto Data
Data (10):
  Assert Dout
  goto Wait2
Wait2 (11):
  goto Idle

```



FSM Logic

```

Dout = S1*S0
NS0 = /S1*S0*Acyc*myAddr*Read + S1*S0
NS1 = /S1*S0 + S1*S0

```

4 4-LUTs

2 LUT Delays

25

Caltech CS184 Winter2005 -- DeHon

FSM Example (Instruction)

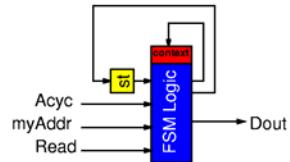
Context 0 (S1=0)

```

Dout = 0
NS0 = /S0*Acyc*myAddr*Read
NS1 = S0

```

3 4-LUTs
1 LUT Delay



26

Context 1 (S1=1)

```

Dout = /S0
NS0 = /S0
NS1 = /S0

```

Caltech CS184 Winter2005 -- DeHon

Full Partitioning Experiment

- Give each state its own context
- Optimize logic in state separately
- Tools
 - mustang, espresso, sis, Chortle
- Use:
 - one-hot encodings for single context
 - smallest/fastest
 - dense for multicontext
 - assume context select needs dense

27

Caltech CS184 Winter2005 -- DeHon

Look at

- Assume stay in context for a number of LUT delays to evaluate logic/next state
- Pick delay from worst-case
- Assume single LUT-delay for context selection?
 - savings of 1 LUT-delay => comparable time
- Count LUTs in worst-case state

28

Caltech CS184 Winter2005 -- DeHon

Full Partition (Area Target)

FSM	States	Single Context		Context per State		Ratio $\frac{\text{LUTs}}{\text{LUTs}}$	Delta Levels
		Levels	λ_{target}	Area (mA^2)	Levels	λ_{target}	Area (mA^2)
bbrora	10	6	25	22.0	1	6	9.5
bbrora	16	4	50	43.9	3	12	24.6
bbrora	6	5	7	1.1	5	1.1	2
bbrora	7	4	14	12.3	1	7	9.4
c10	6	6	83	73.9	2	16	0.77
c14	7	4	58	50.9	1	8	10.8
c15	4	12	28	22.0	1	7	7.8
c16	27	5	80	12	1	8	0.33
c17	8	19	16.7	1	6	8.5	0.51
ds512	15	2	20	17.6	1	7	13.8
dromile	24	2	46	40.4	1	6	16.0
ext	7	21	120	105.4	2	26	61.4
ext	14	7	21	14.4	1	13	3.3
ext	8	5	57	50.9	1	11	15.7
keyb	19	7	112	98.3	4	14	32.0
mc	4	2	8	7.0	1	7	1.10
modulo12	12	6	12	10.5	1	5	8.82
pchoff	48	1	17	25	16.6	16	0.95
pma	24	6	82	72.0	2	16	40.1
s1	20	5	137	120.3	5	26	59.0
s1488	48	6	152	133.5	3	27	122.7
s1c	20	5	72	63.2	7	21	49.6
s208	18	4	38	33.7	7	14	9.78
s27	6	2	5	4.4	1	4	5.1
s386	13	5	42	36.9	2	12	21.8
s420	18	3	40	35.1	1	7	16.8
s510	47	5	54	47.4	1	13	22.2
s5	4	15	5	4	47	4	4.5
s820	25	6	92	80.8	3	30	62.5
sanc	32	7	178	156.3	5	30	98.9
sse	16	4	50	43.9	3	12	24.6
styr	30	7	186	163.3	4	21	65.9
tbk	32	8	345	298.5	6	33	188.8
Average				0.64	3		

29

Caltech CS184 Winter2005 -- DeHon

Full Partition (Delay Target)

FSM	States	Single Context		Context per State		Ratio $\frac{\text{LUTs}}{\text{LUTs}}$	Delta Levels
		Levels	λ_{target}	Area (mA^2)	Levels	λ_{target}	Area (mA^2)
bbrora	10	3	49	35.1	1	6	9.5
bbrora	16	5	85	52.7	2	14	28.4
bbrora	6	2	5	1.9	1	5	0.3
bbrora	7	2	19	16.7	1	7	9.4
cse	16	4	97	85.2	2	15	30.7
dk14	7	3	67	58.8	1	8	10.8
dk15	4	5	37	32.5	1	7	7.8
dk16	27	5	83	79.0	1	8	23.2
dk17	8	2	26	22.8	1	6	8.5
ds632	15	2	29	17.6	1	7	13.8
dromile	24	2	46	40.4	1	6	16.0
ext	7	21	120	105.4	2	26	61.4
ext	14	7	21	14.4	1	13	3.3
ext	8	3	62	54.4	1	11	15.7
keyb	19	4	150	131.7	3	26	59.3
mc	4	2	8	7.0	1	7	7.8
modulo12	12	1	15	14	1	5	8.7
phot	48	4	151	131.0	2	25	59.0
pma	24	4	139	122.0	2	15	40.1
s1	20	4	193	171.2	3	30	70.8
s1488	48	4	183	160.7	2	28	127.2
s1g	20	3	107	93.9	4	30	70.5
s27	6	2	45	38.1	1	5	15.4
s386	13	4	54	47.4	2	12	21.8
s420	18	3	40	35.1	1	7	15.4
s510	47	3	76	66.7	1	13	58.1
s5	4	2	15	13.4	1	4	4.5
s820	25	3	137	130.3	3	30	82.5
sanc	32	4	224	196.7	3	43	141.7
sse	16	3	60	52.7	2	14	28.7
styr	30	5	265	250.2	3	23	72.2
tbk	32	5	510	447.8	4	42	138.4
Average						0.56	1.36

Caltech CS184 Winter2005 -- DeHon

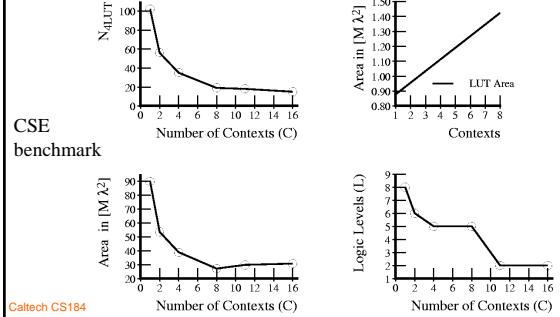
Full Partitioning

- 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)

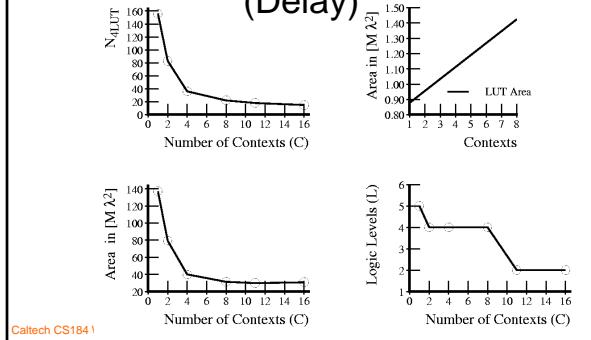
Caltech CS184 Winter2005 -- DeHon

31

Partitioning versus Contexts (Area)



Partitioning versus Contexts (Delay)



Caltech CS184

34

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

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.

35

Caltech CS184 Winter2005 -- DeHon

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	Integral	Memory	FPGA	8-ctx DPGA
				area (Mλ²)	Addr. & Data Organization	area (Mλ²)	area (Mλ²)	area (Mλ²)
bbtfs	6	2	2	0.1	$2^5 \times 5$	0.2	6.1	7.1
dk15	4	3	5	0.3	$2^5 \times 7$	0.3	21.9	10.0
dk17	8	2	3	0.2	$2^5 \times 6$	0.2	16.7	8.5
dk512	15	1	3	0.3	$2^5 \times 7$	0.3	17.6	10.0
mc	4	3	5	0.3	$2^5 \times 7$	0.3	7.0	10.0
modulo12	12	1	1	0.1	$2^5 \times 5$	0.2	10.5	7.1
beecount	7	3	4	0.5	$2^5 \times 7$	0.5	12.3	10.0
dk14	7	3	5	0.5	$2^5 \times 8$	0.6	50.9	11.4
dk16	27	2	3	1.0	$2^5 \times 8$	1.3	70.2	11.4
dontfile	24	2	1	0.7	$2^5 \times 6$	0.9	40.4	8.5
s27	6	4	1	0.5	$2^5 \times 4$	0.6	4.4	5.7
s8	5	4	1	0.4	$2^5 \times 4$	0.6	10.5	5.7
bbara	10	4	2	1.2	$2^5 \times 6$	1.8	21.9	11.4
ex6	8	5	8	3.4	$2^6 \times 11$	3.4	50.0	15.7
ex4	14	6	9	14.0	$2^{10} \times 13$	16.0	18.4	18.5
bbsse	16	7	7	27.0	$2^{11} \times 11$	27.0	43.9	21.4
cse	16	7	7	27.0	$2^{11} \times 11$	27.0	72.9	27.1
tbk	32	6	3	19.7	$2^{11} \times 8$	19.7	298.5	68.4

Memory FSM Compare (large)

FSM	states	ins	outs	Min	Integral	Memory	FPGA	8-ctx DPGA
				area (Mλ²)	Addr. & Data Organization	area (Mλ²)	area (Mλ²)	area (Mλ²)
sse	16	7	7	27.0	$2^{11} \times 11$	27.0	43.9	21.4
s386	13	7	7	22.9	$2^{11} \times 11$	27.0	36.9	18.5
keyb	19	7	2	20.4	$2^{12} \times 7$	34.4	98.3	31.3
planet	48	7	19	184.3	$2^{15} \times 25$	245.8	131.7	54.1
pma	24	8	8	95.8	$2^{13} \times 13$	127.8	72.0	34.2
s1	20	8	6	67.6	$2^{15} \times 11$	108.1	120.3	62.7
s1a	20	8	6	67.6	$2^{15} \times 11$	108.1	63.2	54.1
ex1	20	9	19	294.9	$2^{14} \times 24$	471.9	105.4	55.5
s1488	48	8	19	368.6	$2^{14} \times 25$	491.5	133.5	74.0
styr	30	9	10	276.5	$2^{14} \times 15$	294.9	163.3	57.0
s208	18	11	2	309.7	$2^{16} \times 7$	550.5	33.4	12.8
scnd	32	11	9	1101.0	$2^{16} \times 14$	1101.0	156.3	62.7
s820	25	18	19	188743.7	$2^{25} \times 24$	241591.9	80.8	64.1
s420	18	19	2	79272.3	$2^{24} \times 7$	140928.6	35.1	14.2
s510	47	19	7	384408.9	$2^{26} \times 13$	523449.1	47.4	25.6

Caltech CS184 Winter2005 -- DeHon

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)

39

Caltech CS184 Winter2005 -- DeHon

Control Granularity

Control Granularity

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

41

Caltech CS184 Winter2005 -- DeHon

Consider

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

42

Caltech CS184 Winter2005 -- DeHon

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

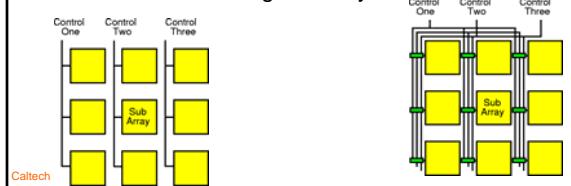
- If FSMs advance orthogonally
 - (really independent control)
 - context depth => product of states
 - for full partition
 - *I.e.* w/ single controller (PC)
 - must create product FSM
 - which may lead to state explosion
 - N FSMs, with S states => S^N product states
- This example:
 - 4 states, 2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

Caltech CS184 Winter2005 -- DeHon

44

Architectural Questions

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



Caltech

46

Architectural Questions

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

Caltech CS184 Winter2005 -- DeHon

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
 - uP requires sequence of operations

49

Caltech CS184 Winter2005 -- DeHon

Architecture Instr. Taxonomy

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

50

Caltech CS184 Winter2005 -- DeHon

Admin

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

51

Caltech CS184 Winter2005 -- DeHon

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

52

Caltech CS184 Winter2005 -- DeHon

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)

53

Caltech CS184 Winter2005 -- DeHon