CS294-7: Reconfigurable Computing

Day 26: Thursday, April 24

Everything Else...

- Admin
- Control

Admin

- Xilinx/Multicontext, April 30, W5pm, Hogan Room
- Projects Due in 2 weeks
- Signup for Project report slots
  - 10 Minute talk slots
  - Three periods
    (Class time T and R + T afternoon 5-6:30pm)

Control

Control $\rightarrow$ That point where the data affects the instruction stream (instruction selection).

Viewpoint: can have instruction stream sequence without control

Typically manifest as:

- **branching**
- **data dependent state transition**
- **data dependent operation selection**

Terminology (from Day 3)

Primitive Instruction (*pinst*): The collection of bits which tells a bit-processing element how to perform.

includes:

- **selection of compute operation**
- **input sources in space (interconnect)**
- **input sources in time (retiming)**

Configuration Context: collection of all the bits (all the *pinsts*) which describe the behavior of a general-purpose machine on one operation cycle.

Why?

Why do we have (want) control in our architecture?

Consider

- static interconnect
- static (data independent) instruction sequencing

Generalization

We must *generalize* our design to handle all potential inputs or computing scenarios.

However, what we need to do for any given input may be much smaller than the general case.

Two "control" options:

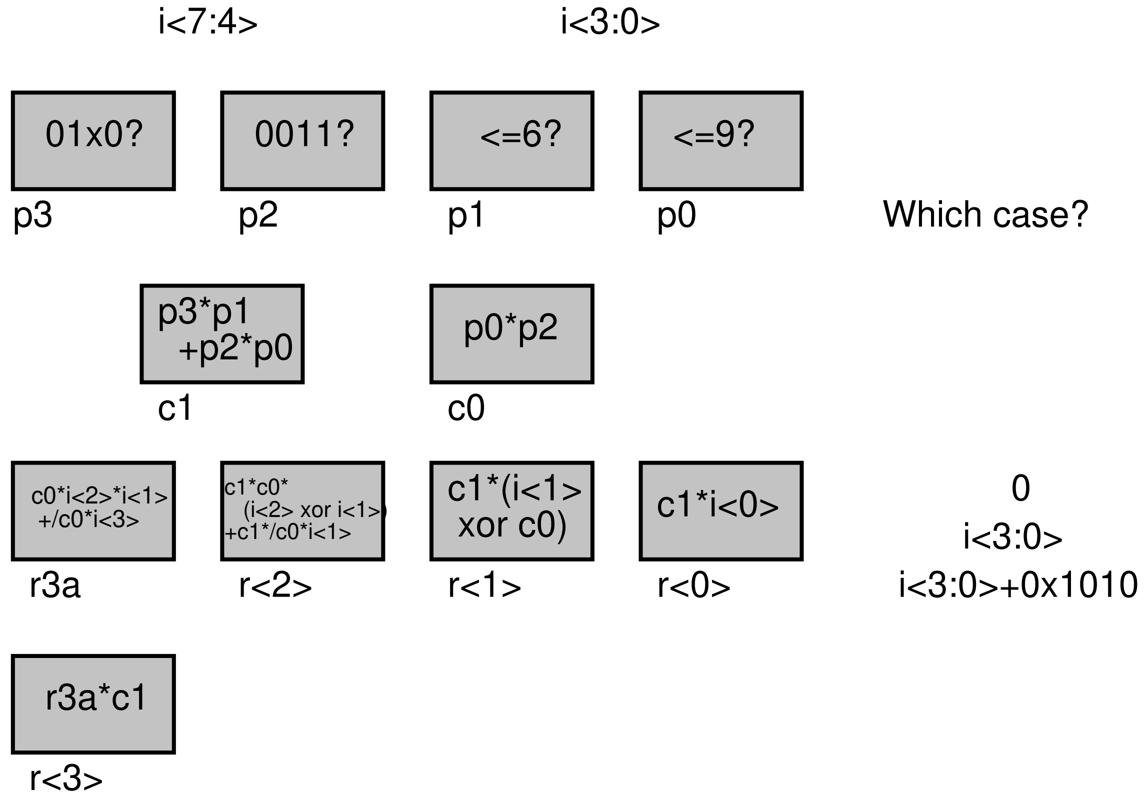
- build all options into spatial computing structure provide (control?) inputs to select operation
- provide different instruction (sequence) for each option; selection comes from choice of instruction(s)

ASCII HEX→Binary Conversion

e.g. **'A'** $\equiv$ **0x41** $\Rightarrow$ **0x0A (0d10)**

```
if (c >= 0x30 && c <= 0x39)
    res = c-0x30;
else if (c >= 0x40 && c <= 0x46)
    res = c - 0x40 + 10;
else if (c >= 0x60 && c <= 0x66)
    res = c - 0x60 + 10;
else
    res = 0;
```

Local Control



11 4-LUTs, depth 4

Local Control

Once we start rolling this kind of control into the computation, counting LUTs doesn't really tell us the cycle-by-cycle computing needs:

LUTs used $>$ LUT evaluations produced

e.g. previous example only needed 4 LUTs for each case; 5 required for unification

Local Control

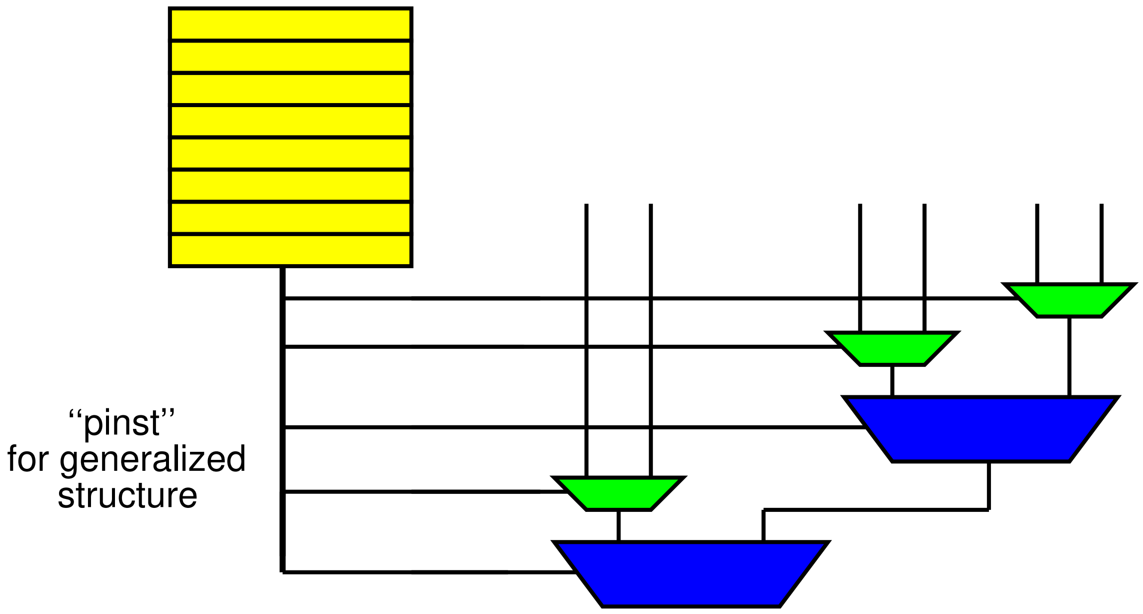
Local control can only exploit resources which data has control over in the architecture

e.g. if we attempt to unify: a+b+c+d and e+f+g+h

Requires: 3 4-LUTs
(5 if naively mux inputs and share OR)

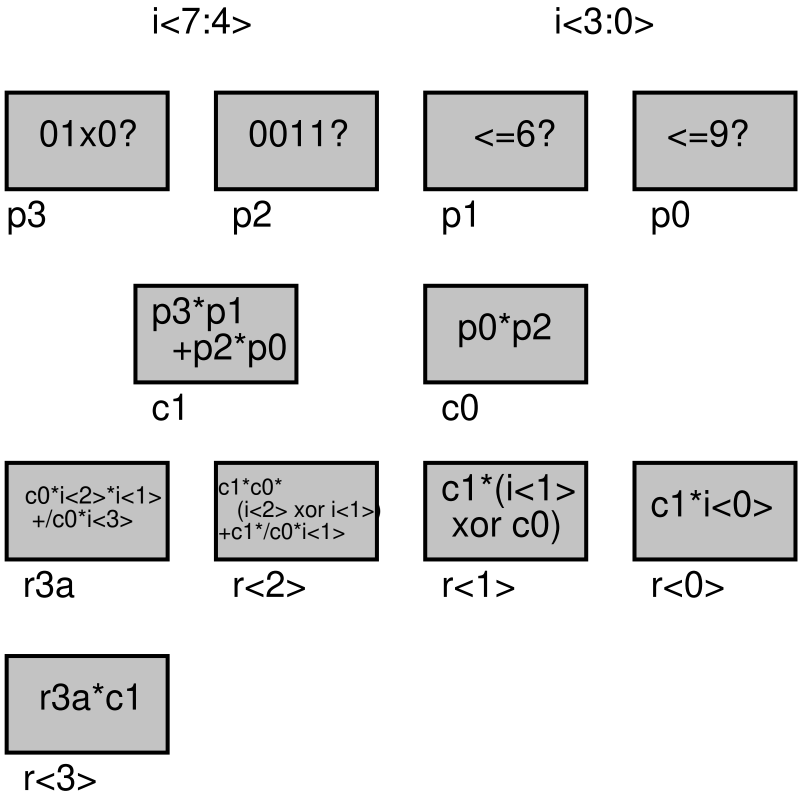
Building Interpreters / Task Description Hierarchy

These "local control" signals are sometimes packaged together into a "pinst" for an interpreter one level up in our design hierarchy



(This may cause some amount of confusion between the two cases)

Instruction Selection Control



No Instruction Control

p3	p2	p1	p0
		c1	c0
r3a	r2	r1	r0
r3			

Instruction Control

00	p3	p2	p1	p0	0	1
01				c0	1	c1
10	0	0	0	0	0	0
11	r3	r2	r1	r0	0	0

No control: 4 4-LUTs, depth 4, 4 (3?) contexts
Control: 6 4-LUTs, depth 3, 4 contexts

Instruction Selection Control

Architectural Question: How many pinsts/controller?

- Orthogonal control

If have s independent things
(each advancing/branching separately),
each of which can have N instruction slots,
the total number of unique (possible?),
composite instructions is N^s.

With single stream device,
now need N^s times more memory than is possible
when separately controlled.

Architecture Instruction Taxonomy

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

Instruction Selection Control

Architectural Question: How many levels/points of translation?

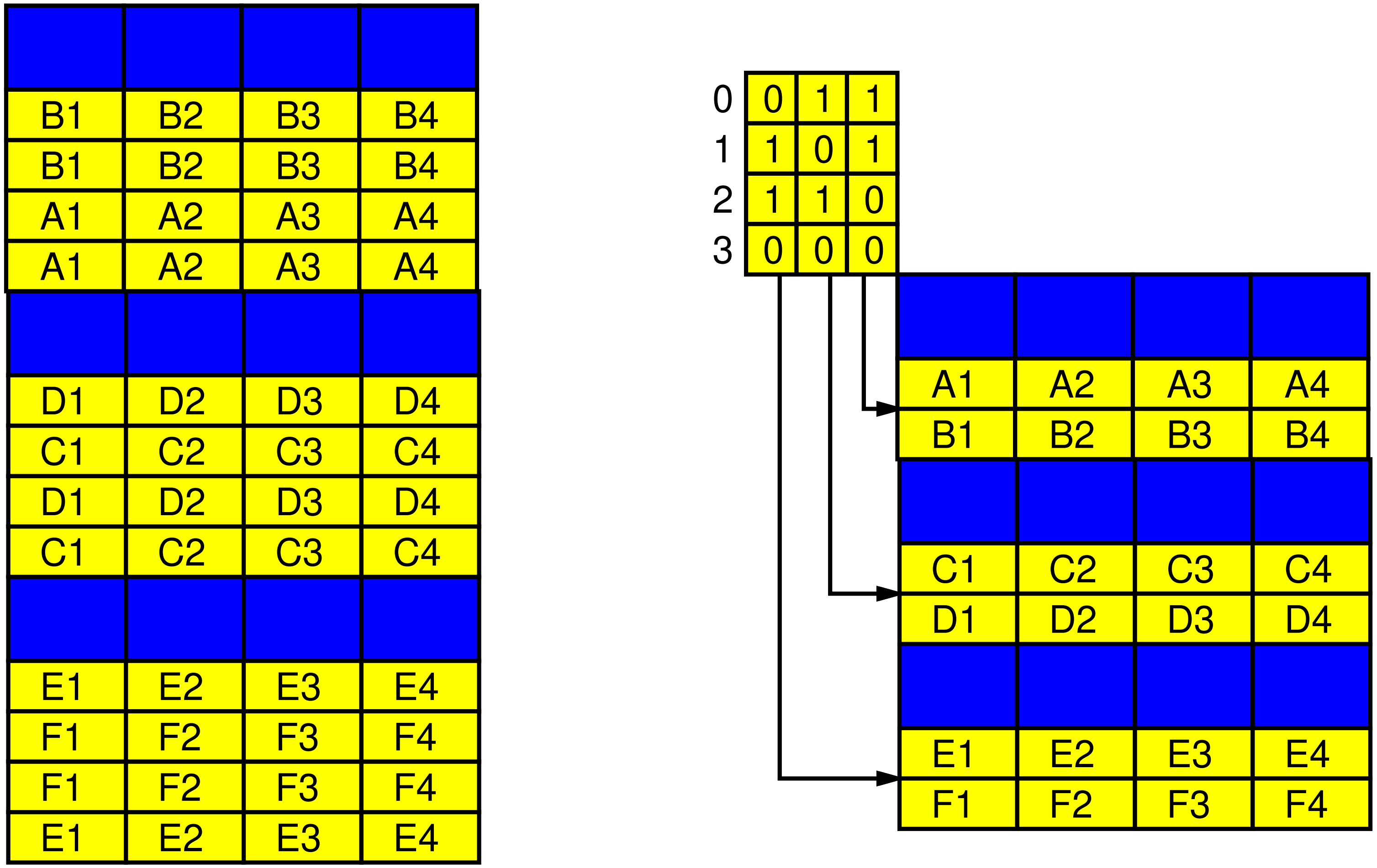
- Instruction redundancy

Similar effect to orthogonal control.

Collection of n_s pinsts:

- potentially $2^{\left(n_s \cdot |\textbf{pinst}|\right)}$ instructions
- typically only want small subset in a task – n_{op}
- however, may be many such subsets, which combine
- total number of "distinct" instructions: $n_{op1} \cdot n_{op2} \cdots n_{opk}$
- translation to smaller context ideas at group level

Translation



Control Formulation

How do we formulate all this?

- local v/s instruction
- costs of controller
- overhead given too much (what's wasted)
- how best compensate when too little (how inefficient)

Compare formulation for:

- datapath width
- context depth
- interconnect richness

Finite State Machines

FSMs – canonical "control" structure
captures many of these properties

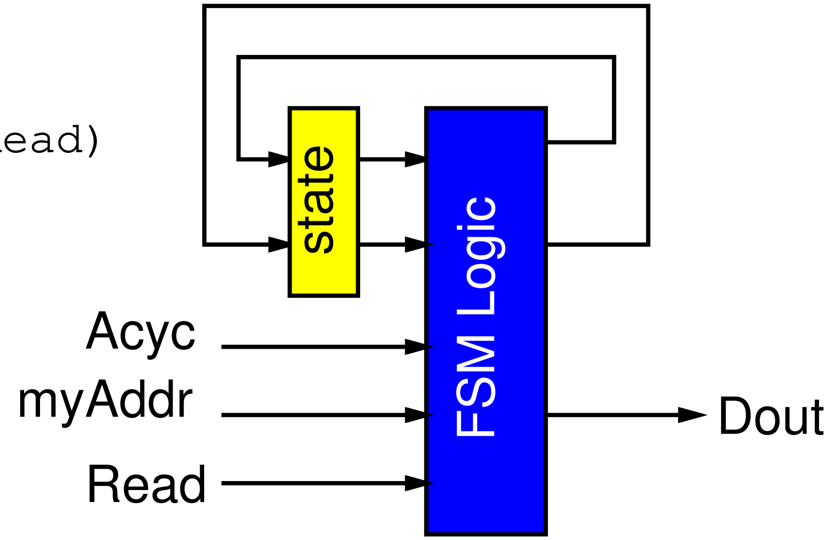
- can implement with deep multicontext (microcode)
- can implement as multilevel (unifying state logic)

Allows us to explore some of the "local control" unification versus "instruction control" tradeoffs.

FSM Example

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
```

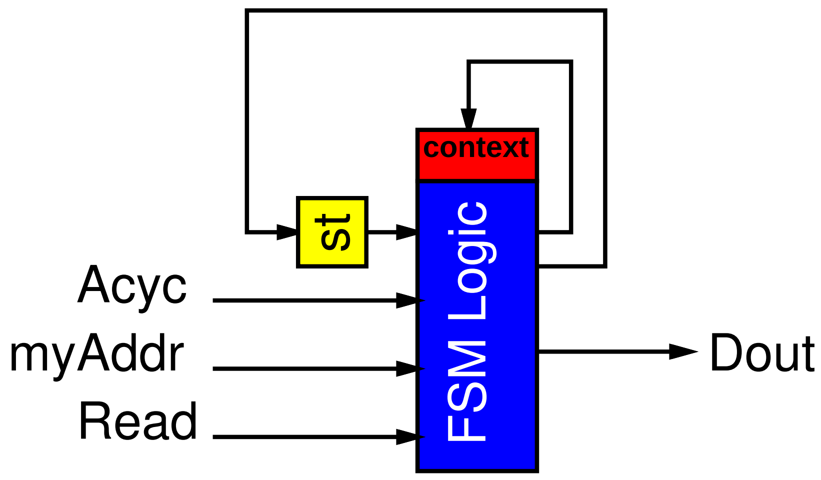
Multicontext FSM Example

Context 0 `(S1=0)`

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

Context 1 `(S1=1)`

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



Full Partitioning Experiment

- Give each state its own context
- Optimize logic in state separately

Tools:

- mustang
- espresso, sis
- Chortle

N.B. one-hot encodings for single context; dense for multicontext

Full Partition -- Area Target

FSM	States	Single Context Levels	Single Context N_{4LUT}	Single Context **Area** (Mλ^2)	Context per State Levels	Context per State N_{4LUT}	Context per State **Area** (Mλ^2)	**Ratio** $\frac{A_{multi}}{A_{single}}$	**Delta Levels**
bbara	10	6	25	22.0	1	6	9.5	0.43	5
bbsse	16	4	50	43.9	3	12	24.6	0.56	1
bbtas	6	3	7	6.1	1	5	6.34	1.0	2
beecount	7	4	14	12.3	1	7	9.4	0.77	3
cse	16	6	83	72.9	2	15	30.7	0.42	4
dk14	7	4	58	50.9	1	8	10.8	0.21	3
dk15	4	12	25	22.0	1	7	7.8	0.35	11
dk16	27	5	80	70.2	1	8	23.2	0.33	4
dk17	8	6	19	16.7	1	6	8.5	0.51	5
dk512	15	2	20	17.6	1	7	13.8	0.79	1
donfile	24	2	46	40.4	1	6	16.0	0.40	1
ex1	20	7	120	105.4	2	26	61.4	0.58	5
ex4	14	7	21	18.4	1	13	24.6	1.33	6
ex6	8	5	57	50.0	1	11	15.7	0.31	4
keyb	19	7	112	98.3	4	14	32.0	0.32	3
mc	4	2	8	7.0	1	7	7.8	1.10	1
modulo12	12	6	12	10.5	1	5	8.7	0.82	5
planet	48	6	150	131.7	1	25	113.6	0.86	5
pma	24	6	82	72.0	2	15	40.1	0.56	4
s1	20	5	137	120.3	5	25	59.0	0.49	0
s1488	48	6	152	133.5	3	27	122.7	0.92	3
s1a	20	5	72	63.2	7	21	49.6	0.78	-2
s208	18	4	38	33.4	1	7	15.4	0.46	3
s27	6	2	5	4.4	1	4	5.1	1.20	1
s386	13	5	42	36.9	2	12	21.8	0.59	3
s420	18	3	40	35.1	1	7	15.4	0.44	2
s510	47	5	54	47.4	1	13	58.1	1.22	4
s8	5	4	12	10.5	1	4	4.7	0.45	3
s820	25	6	92	80.8	3	30	82.5	1.02	3
sand	32	7	178	156.3	5	30	98.9	0.63	2
sse	16	4	50	43.9	3	12	24.6	0.56	1
styr	30	7	186	163.3	4	21	65.9	0.40	3
tbk	32	8	340	298.5	6	33	108.8	0.36	2
Average								0.64	3

Full Partition -- Delay Target

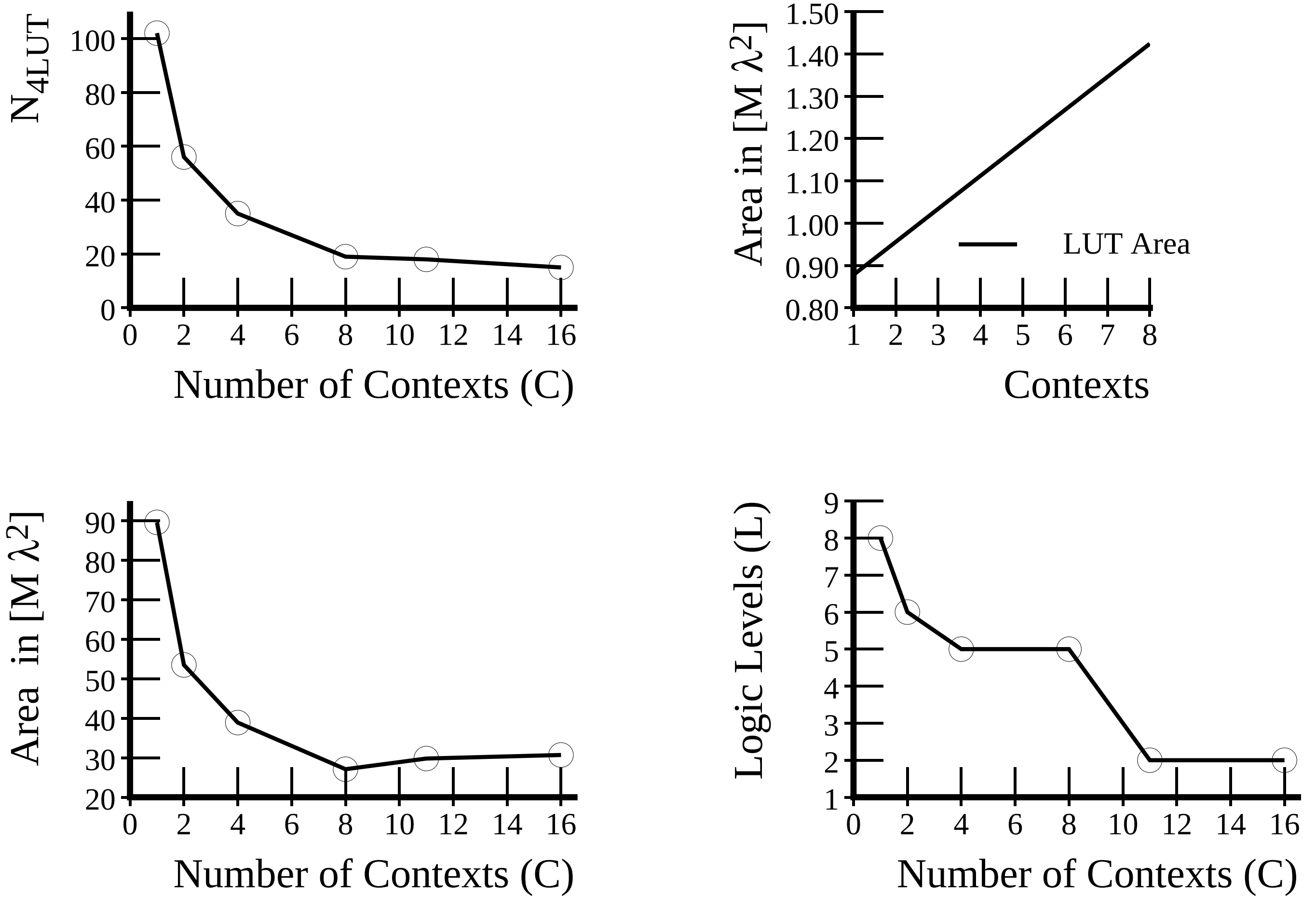
FSM	States	Single Context: Levels	Single Context: N_{4LUT}	Single Context: **Area** (Mλ^2)	Context per State: Levels	Context per State: N_{4LUT}	Context per State: **Area** (Mλ^2)	**Ratio** $\frac{A_{multi}}{A_{single}}$	**Delta Levels**
bbara	10	3	40	35.1	1	6	9.5	0.27	2
bbsse	16	3	60	52.7	2	14	28.7	0.54	1
bbtas	6	2	9	7.9	1	5	6.3	0.80	1
beecount	7	2	19	16.7	1	7	9.4	0.57	1
cse	16	4	97	85.2	2	15	30.7	0.36	2
dk14	7	3	67	58.8	1	8	10.8	0.18	2
dk15	4	3	37	32.5	1	7	7.8	0.24	2
dk16	27	3	83	72.9	1	8	23.2	0.32	2
dk17	8	2	26	22.8	1	6	8.5	0.37	1
dk512	15	2	20	17.6	1	7	13.8	0.79	1
donfile	24	2	46	40.4	1	6	16.0	0.40	1
ex1	20	4	151	132.6	2	26	61.4	0.46	2
ex4	14	2	25	22.0	1	13	24.6	1.12	1
ex6	8	3	62	54.4	1	11	15.7	0.29	2
keyb	19	4	150	131.7	3	26	59.3	0.45	1
mc	4	2	8	7.0	1	7	7.8	1.10	1
modulo12	12	1	13	11.4	1	5	8.7	0.76	0
planet	48	4	172	151.0	1	25	113.6	0.75	3
pma	24	4	139	122.0	2	15	40.1	0.33	2
s1	20	4	195	171.2	3	30	70.8	0.41	1
s1488	48	4	183	160.7	2	28	127.2	0.79	2
s1a	20	3	107	93.9	4	30	70.8	0.75	-1
s208	18	3	40	35.1	1	7	15.4	0.44	2
s27	6	2	5	4.4	1	4	5.0	1.16	1
s386	13	4	54	47.4	2	12	21.8	0.46	2
s420	18	3	40	35.1	1	7	15.4	0.44	2
s510	47	3	76	66.7	1	13	58.1	0.87	2
s8	5	2	13	11.4	1	4	4.8	0.42	1
s820	25	3	137	120.3	3	30	82.5	0.69	0
sand	32	4	224	196.7	3	43	141.7	0.72	1
sse	16	3	60	52.7	2	14	28.7	0.54	1
styr	30	5	285	250.2	3	23	72.2	0.29	2
tbk	32	5	510	447.8	4	42	138.4	0.31	1
Average								0.56	1.36

Full Partitioning

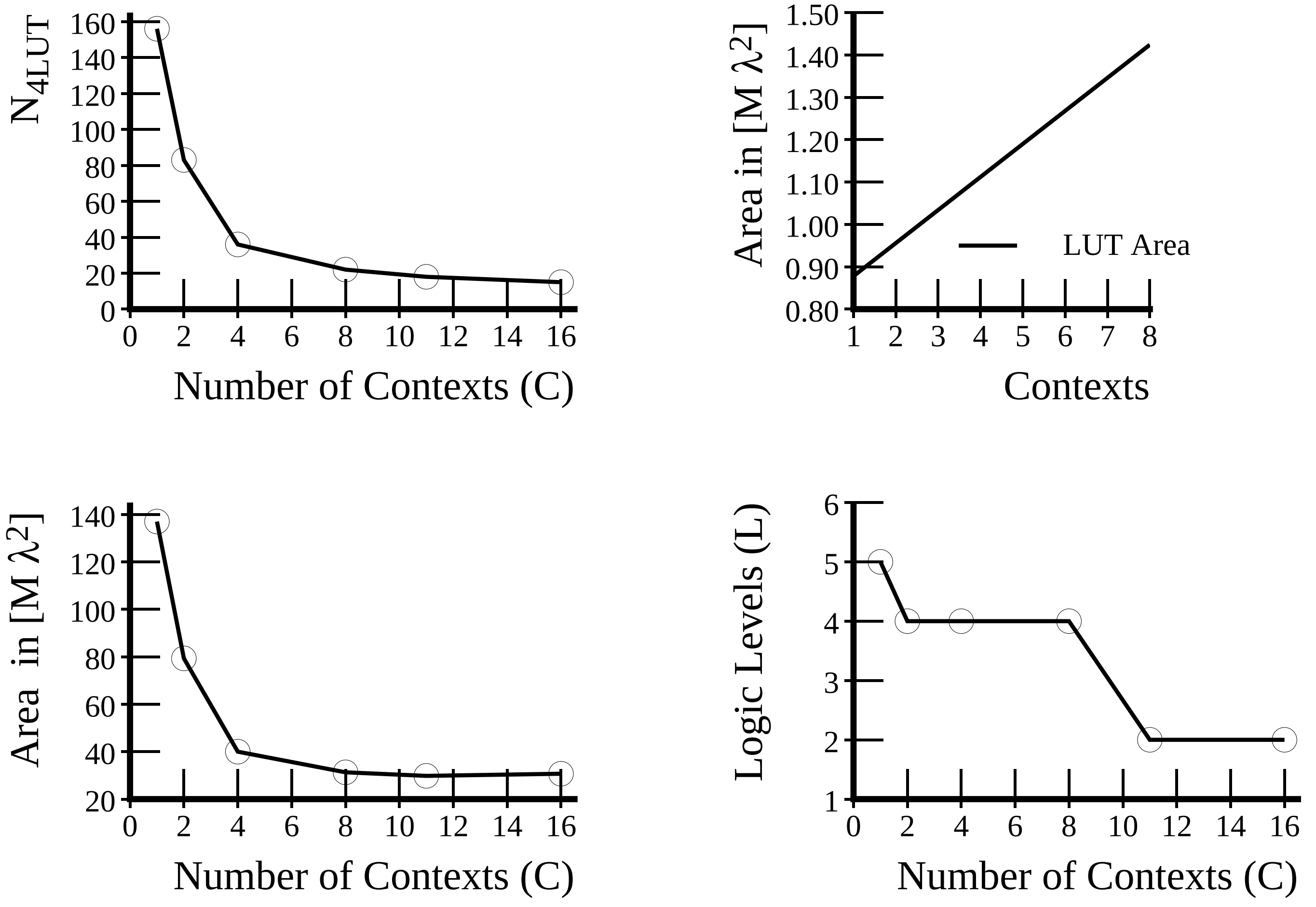
Full partition may not be optimal area point.

e.g. For illustration example,
no reduction in area or time beyond
2-context implementation.

Partitioning versus Contexts (Area)



Partitioning versus Contexts (Delay)



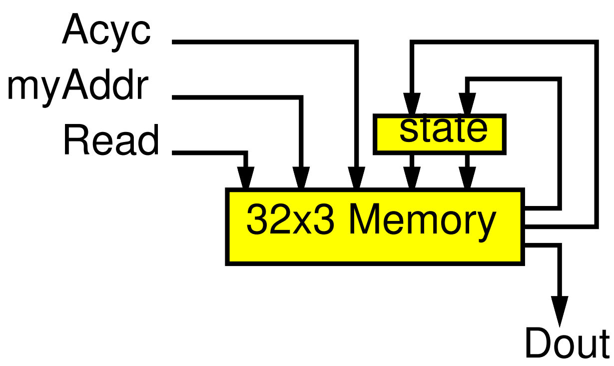
Partition to Fixed Context Number

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. **partitioning along mustang state bits**

Compare versus Memory

Note that we can implement FSM with a memory (ROM).



Compare "optimal" memory implementation
(memory with size matching FSM requirements)

Memory FSM Compare

FSM	states	ins	outs	**Min area** ($M\lambda^2$)	Integral Addr. & Data Organization	**Memory area** ($M\lambda^2$)	**FPGA area** ($M\lambda^2$)	**8-ctx DPGA area** ($M\lambda^2$)
bbtas	6	2	2	0.1	$2^5\times5$	0.2	6.1	7.1
dk15	4	3	5	0.3	$2^5\times7$	0.3	21.9	10.0
dk17	8	2	3	0.2	$2^5\times6$	0.2	16.7	8.5
dk512	15	1	3	0.3	$2^5\times7$	0.3	17.6	10.0
mc	4	3	5	0.3	$2^5\times7$	0.3	7.0	10.0
modulo12	12	1	1	0.1	$2^5\times5$	0.2	10.5	7.1
beecount	7	3	4	0.5	$2^6\times7$	0.5	12.3	10.0
dk14	7	3	5	0.5	$2^6\times8$	0.6	50.9	11.4
dk16	27	2	3	1.0	$2^7\times8$	1.3	70.2	11.4
donfile	24	2	1	0.7	$2^7\times6$	0.9	40.4	8.5
s27	6	4	1	0.5	$2^7\times4$	0.6	4.4	5.7
s8	5	4	1	0.4	$2^7\times4$	0.6	10.5	5.7
bbara	10	4	2	1.2	$2^8\times6$	1.8	21.9	11.4
ex6	8	5	8	3.4	$2^8\times11$	3.4	50.0	15.7
ex4	14	6	9	14.0	$2^{10}\times13$	16.0	18.4	18.5
bbsse	16	7	7	27.0	$2^{11}\times11$	27.0	43.9	21.4
cse	16	7	7	27.0	$2^{11}\times11$	27.0	72.9	27.1
tbk	32	6	3	19.7	$2^{11}\times8$	19.7	298.5	68.4
sse	16	7	7	27.0	$2^{11}\times11$	27.0	43.9	21.4
s386	13	7	7	22.9	$2^{11}\times11$	27.0	36.9	18.5
keyb	19	7	2	20.4	$2^{12}\times7$	34.4	98.3	31.3
planet	48	7	19	184.3	$2^{13}\times25$	245.8	131.7	54.1
pma	24	8	8	95.8	$2^{13}\times13$	127.8	72.0	34.2
s1	20	8	6	67.6	$2^{13}\times11$	108.1	120.3	62.7
s1a	20	8	6	67.6	$2^{13}\times11$	108.1	63.2	54.1
ex1	20	9	19	294.9	$2^{14}\times24$	471.9	105.4	55.5
s1488	48	8	19	368.6	$2^{14}\times25$	491.5	133.5	74.0
styr	30	9	10	276.5	$2^{14}\times15$	294.9	163.3	57.0
s208	18	11	2	309.7	$2^{16}\times7$	550.5	33.4	12.8
sand	32	11	9	1101.0	$2^{16}\times14$	1101.0	156.3	62.7
s820	25	18	19	188743.7	$2^{23}\times24$	241591.9	80.8	64.1
s420	18	19	2	79272.3	$2^{24}\times7$	140928.6	35.1	14.2
s510	47	19	7	384408.9	$2^{25}\times13$	523449.1	47.4	25.6

Summary

- Control – data effects instructions
- allows us to reduce instantaneous work requirement
- local control
- instruction selection control
- Moderate context (*e.g.* 8) FSM – exploits both
  better than either extreme
  (single context, n-context)