Panoptic Dynamic Voltage Scaling: A Fine-Grained Dynamic Voltage Scaling Framework for Energy Scalable CMOS Design

Mateja Putic, Liang Di,
Dr. Benton H. Calhoun, and Dr. John Lach
University of Virginia

2

Panoptic > including everything visible in one view,

2 investigate the merits of fine-grained header switches,

3 major contribution is measured overhead from chip

The CMOS Energy Equation

$$E = \underbrace{\alpha C V_{DD}^2}_{\text{Active Energy}} + \underbrace{I_{static} V_{DD} t}_{\text{Static Energy}}$$

common characteristic of signal processing applications > variable incoming workloads 2 take advantage of these by scaling processing rate to save energy 3 if energy avail. changes, extend lifetime of appl. by degrading performance instead of shutdown

DVS Background

$$r_{opt} = \frac{t_{max}}{t_x} r_{max}$$

Optimal energy is achieved when processing rate == incoming workload

4

1 systems w/ dynamic workload >optimal energy achieved when processing rate == incoming workload 2 work processed as slow as possible while still meeting deadline 3 efficiency of DVS arch. evaluated by range of proc. rates achieved and ovh. of transitions

Overview of Presentation

- Background of DVS Architectures
- Single-Function, Single Rate
- Single-Function, Multi-Rate

5

DVS Bkg. > DVS archs. for SFSR, SFMR, MF and how each of these process dynamic workloads

Overview of Presentation

6

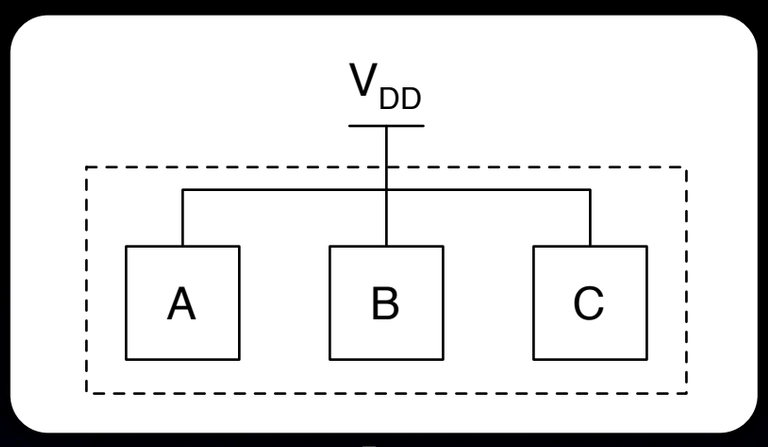
PDVS overheads and measured quantities, break-even point calculation, test chip & measurements, results, conclusions

Overview of Presentation

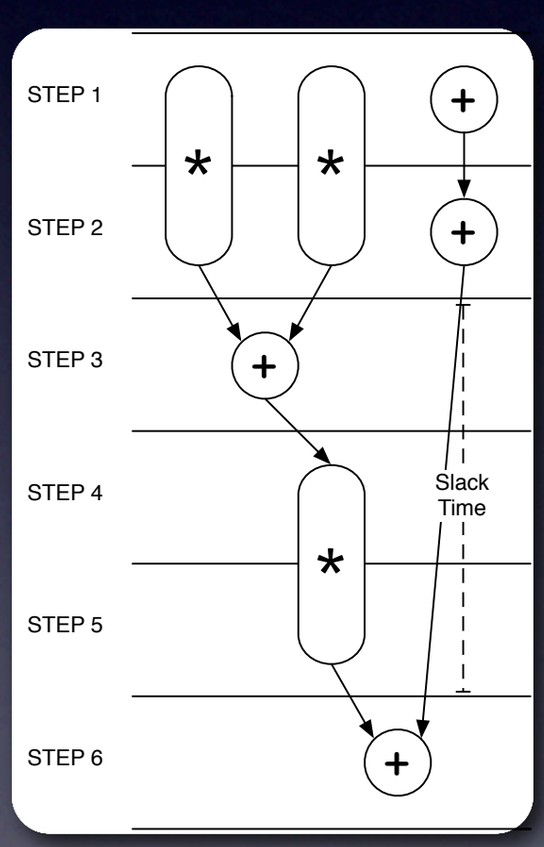
- PDVS overheads and measured quantities
- Break-even Point Calculation
- Test Chip and Measurements
- Results
- Conclusions

6

PDVS overheads and measured quantities, break-even point calculation, test chip & measurements, results, conclusions



Single V_{DD} Architecture



Schedule of Execution

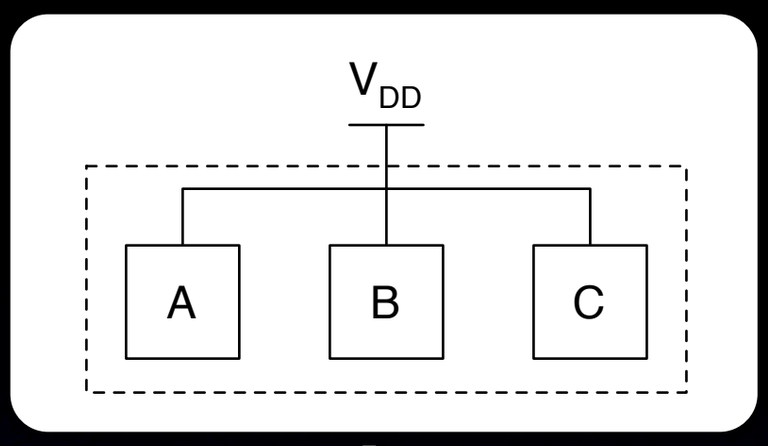
Workload: 0.4
Processing Rate: 1.0

7

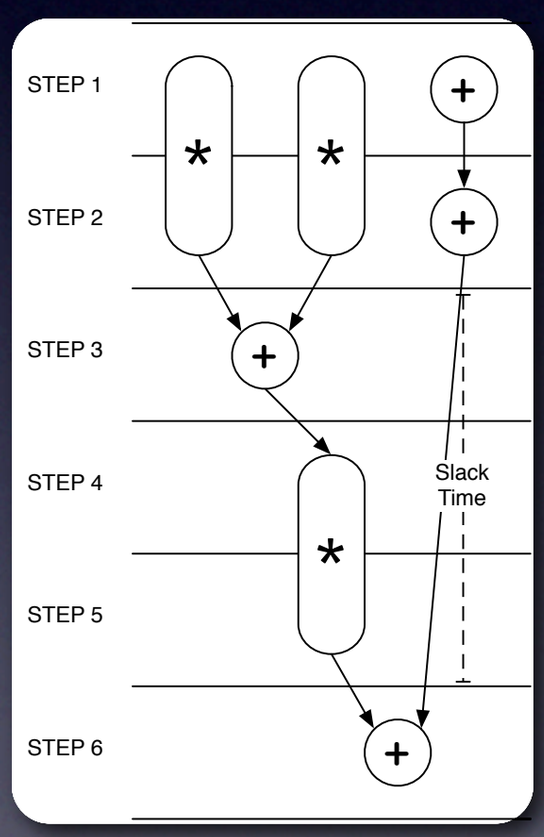
1 SingleVDD, executing single function at single processing rate

2 Example algorithm is static schedule with slack

3 executed at maximum rate, idle until deadline, linear energy savings only

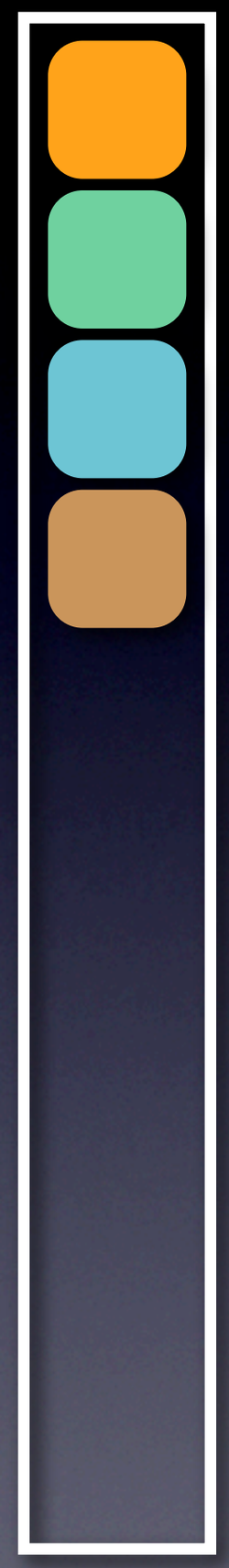


Single V_{DD} Architecture



Schedule of Execution

Workload: 0.4
Processing Rate: 1.0

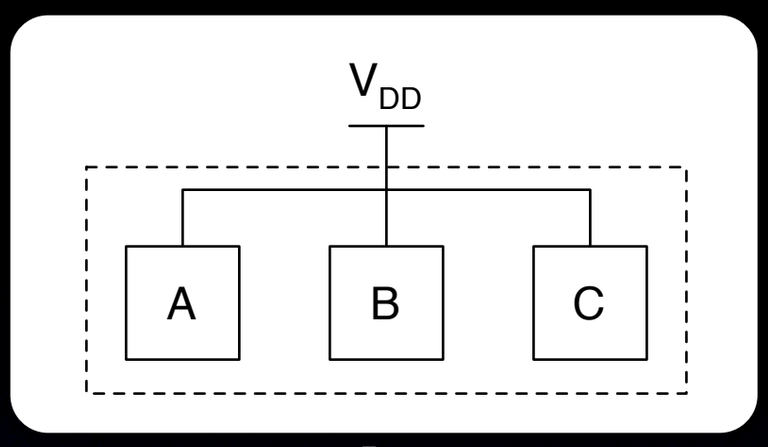


7

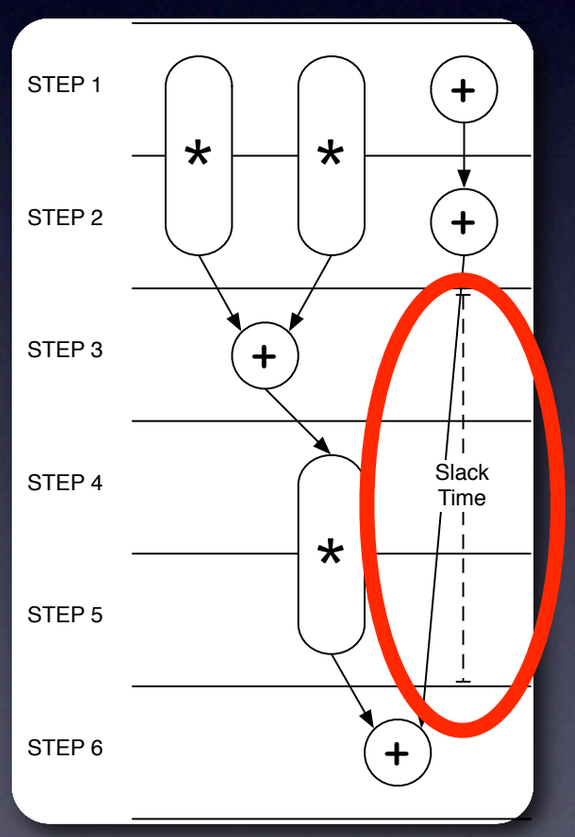
1 SingleVDD, executing single function at single processing rate

2 Example algorithm is static schedule with slack

3 executed at maximum rate, idle until deadline, linear energy savings only

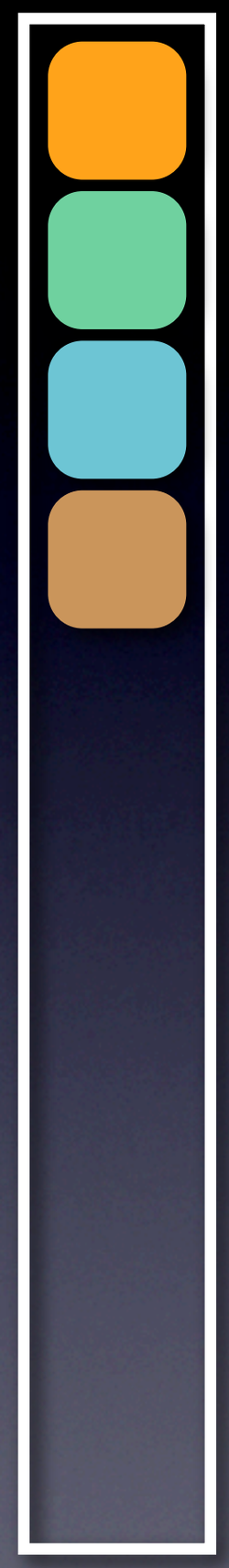


Single V_{DD} Architecture



Schedule of Execution

Workload: 0.4
Processing Rate: 1.0

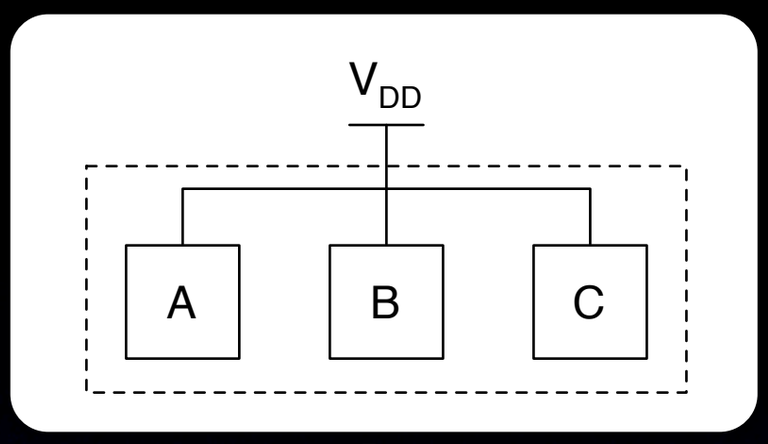


7

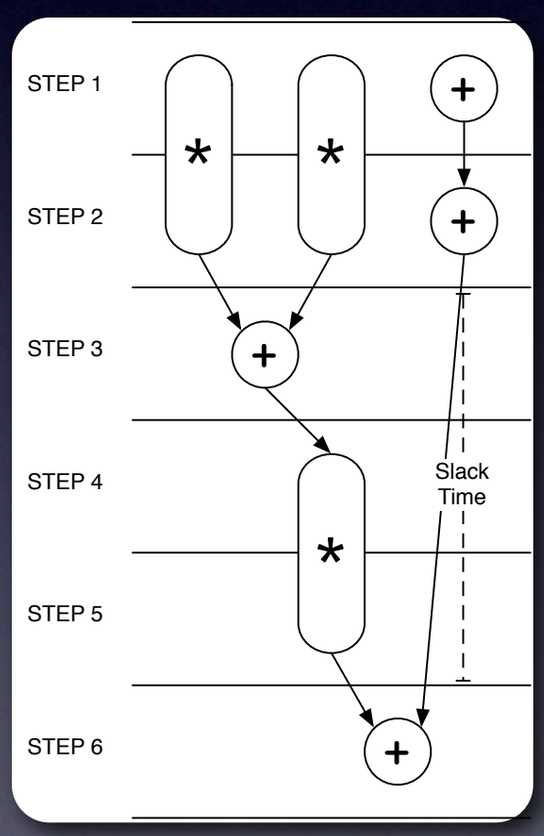
1 SingleVDD, executing single function at single processing rate

2 Example algorithm is static schedule with slack

3 executed at maximum rate, idle until deadline, linear energy savings only



Single V_{DD} Architecture



Schedule of Execution

Workload: 0.4
Processing Rate: 1.0

Workload executed at maximum rate

Idle until deadline is reached

7

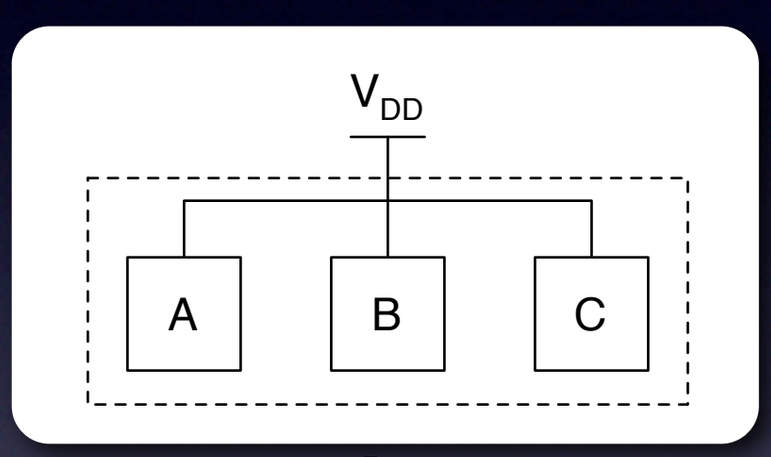
1 SingleVDD, executing single function at single processing rate

2 Example algorithm is static schedule with slack

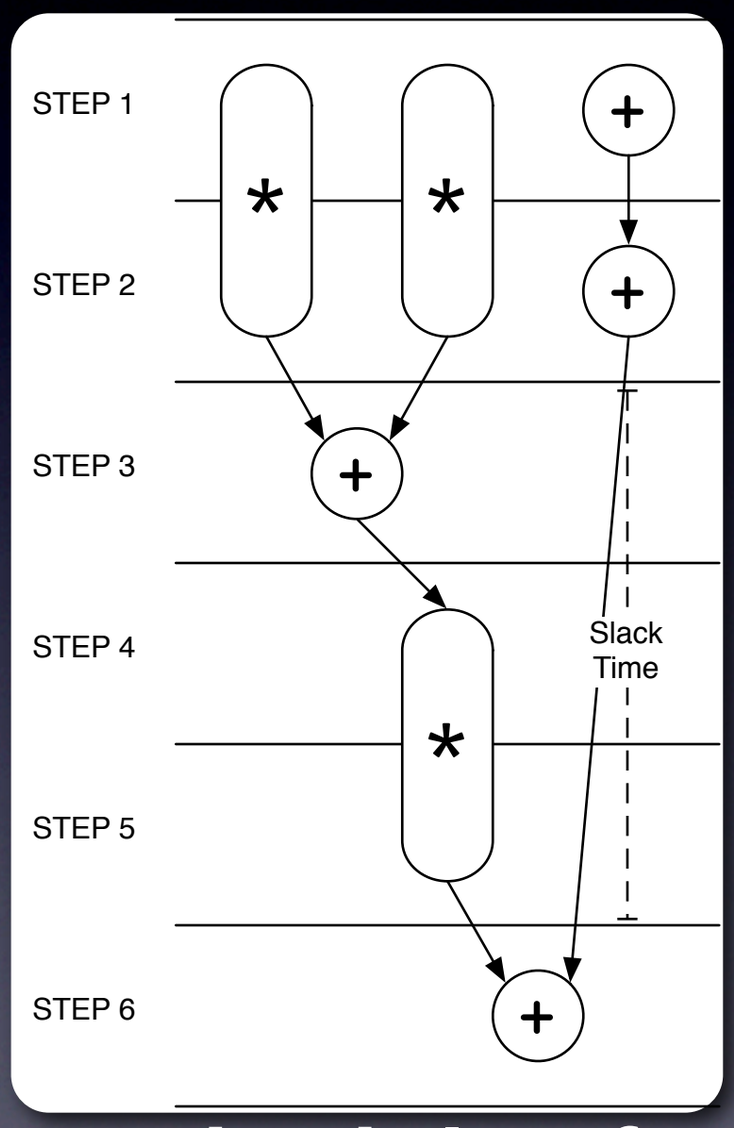
3 executed at maximum rate, idle until deadline, linear energy savings only

DVS Architectures

Single Function Single Rate



Single V_{DD} Architecture



Schedule of Execution

8

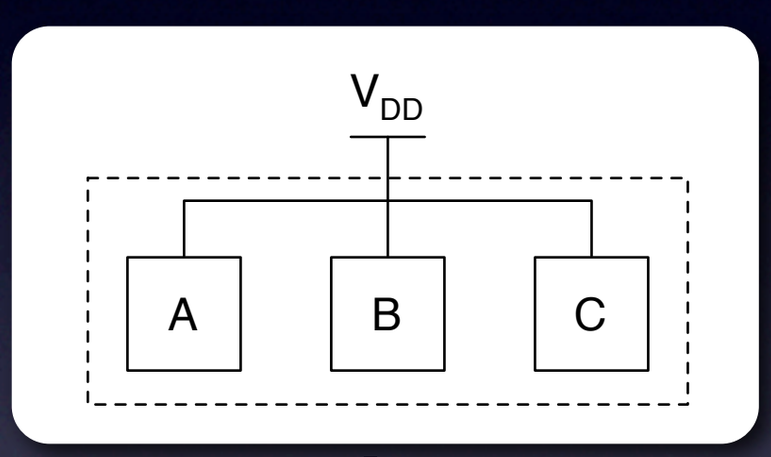
1 Parallel execution architecture

2 Slack exists within algorithm

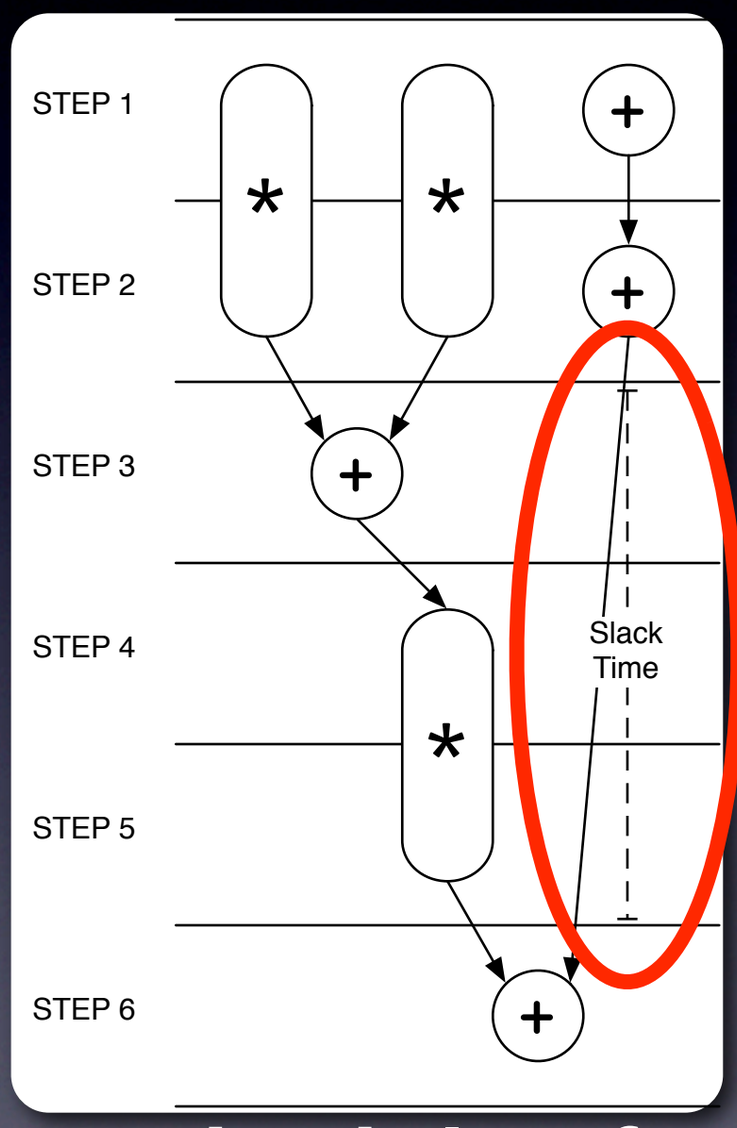
3 Add spatial granularity > ability to choose processing rates of individual operations within the algorithm

DVS Architectures

Single Function Single Rate



Single V_{DD} Architecture



Schedule of Execution

8

1 Parallel execution architecture

2 Slack exists within algorithm

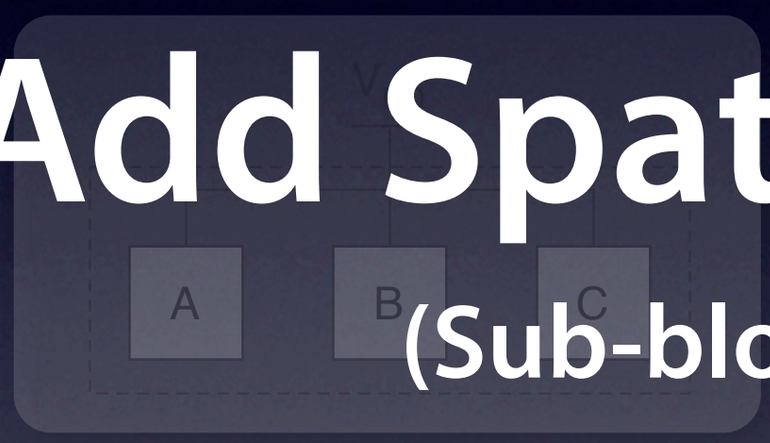
3 Add spatial granularity > ability to choose processing rates of individual operations within the algorithm

DVS Architectures

Single Function Single Rate

Add Spatial Granularity

(Sub-block voltage control)



Single V_{DD} Architecture

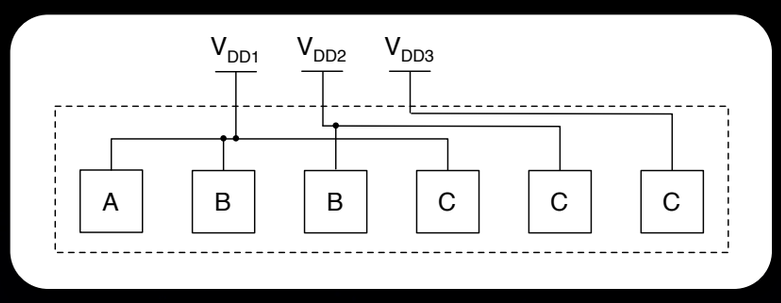
Schedule of Execution

8

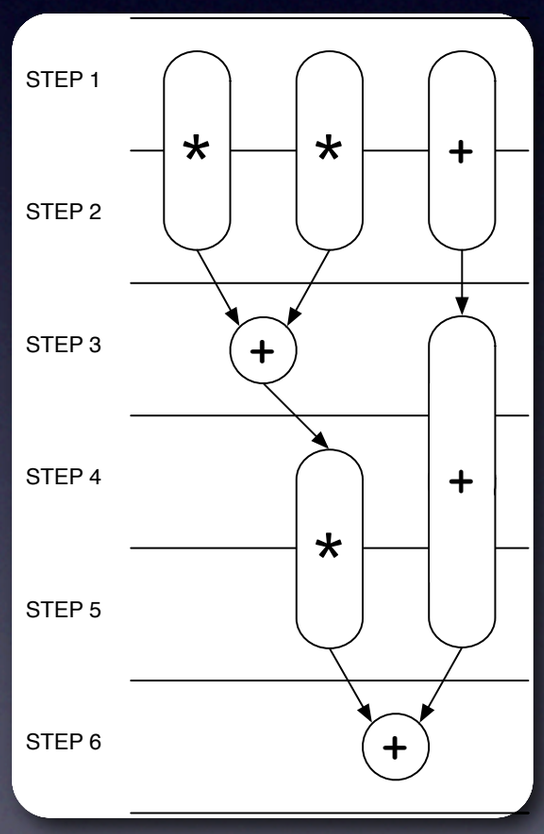
1 Parallel execution architecture

2 Slack exists within algorithm

3 Add spatial granularity > ability to choose processing rates of individual operations within the algorithm



Multi-V_{DD} Architecture



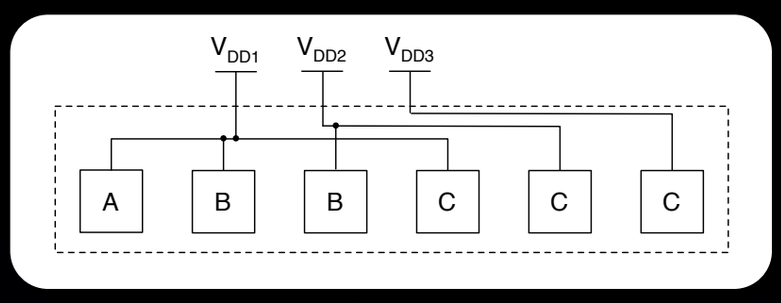
Schedule of Execution

Workload: 0.4
Processing Rate: 1.0

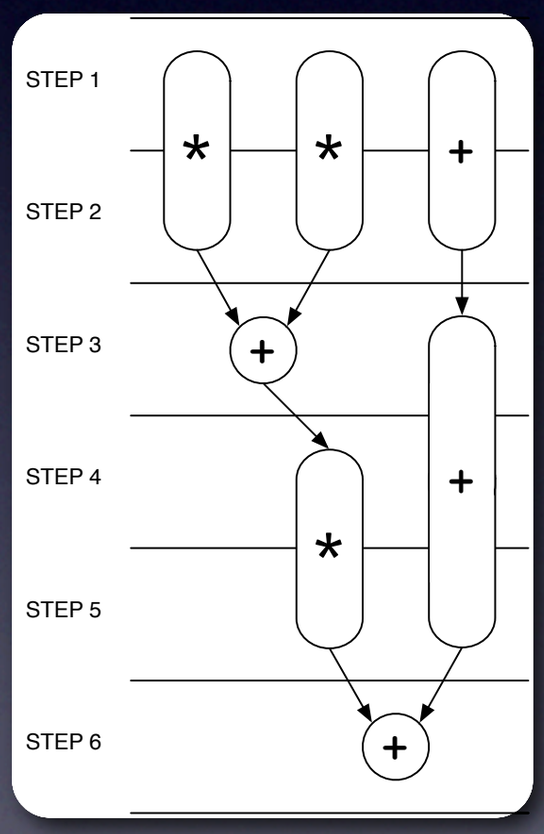
9

1 energy savings is still linear, only available processing rate still maximum

2 but energy is lower b/c each iteration takes less energy due to slack fill

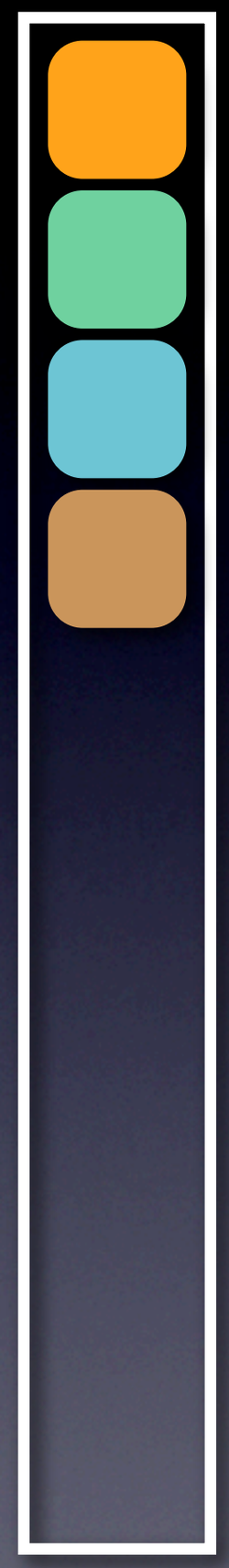


Multi-V_{DD} Architecture



Schedule of Execution

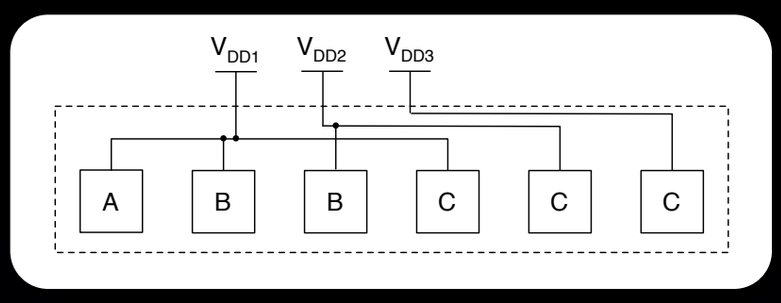
Workload: 0.4
Processing Rate: 1.0



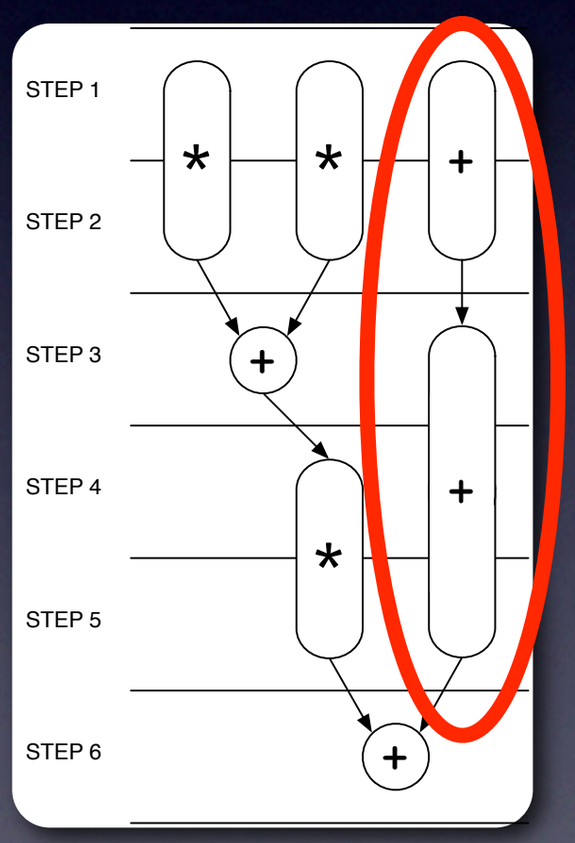
9

1 energy savings is still linear, only available processing rate still maximum

2 but energy is lower b/c each iteration takes less energy due to slack fill



Multi-V_{DD} Architecture



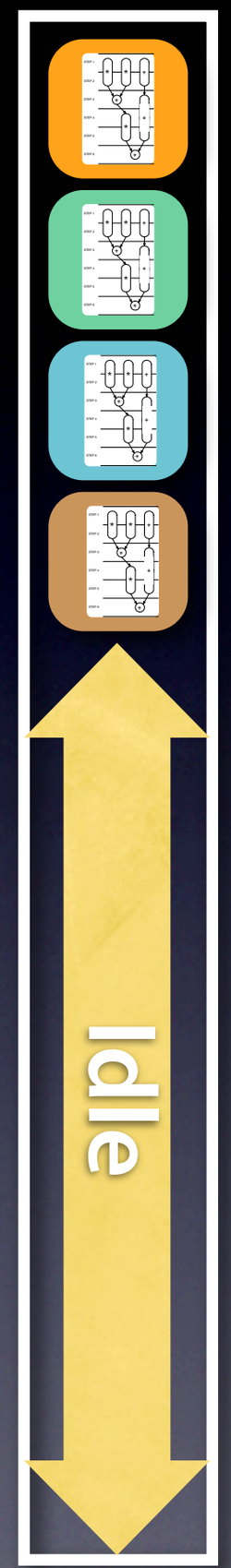
Schedule of Execution

Workload: 0.4
Processing Rate: 1.0

Workload executed at maximum rate

Idle until deadline is reached

With sub-block energy savings



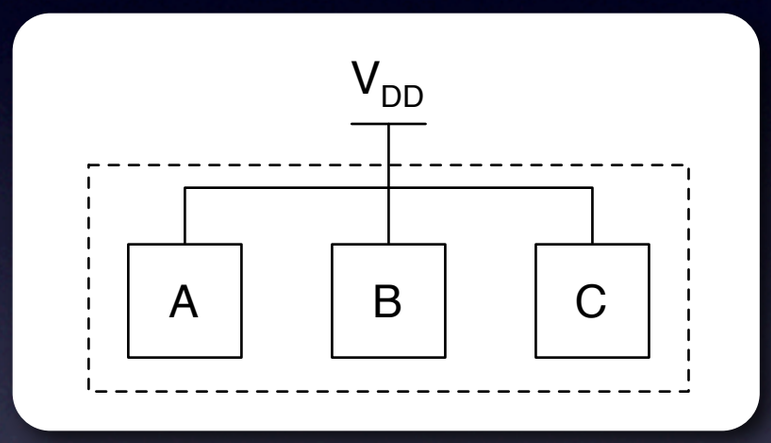
9

1 energy savings is still linear, only available processing rate still maximum

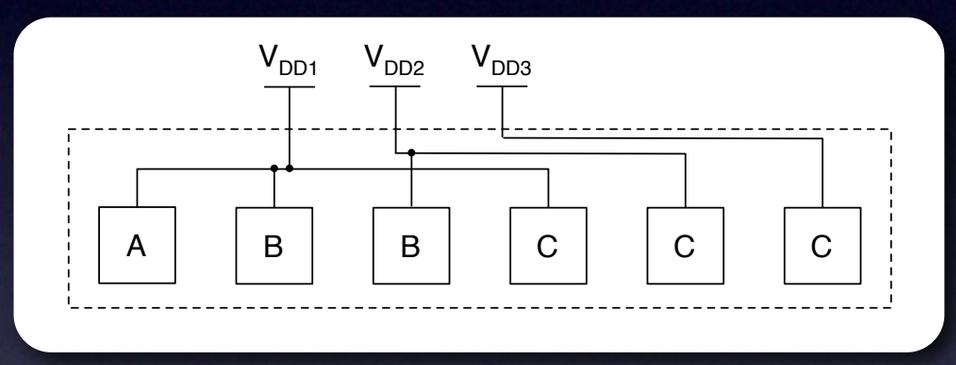
2 but energy is lower b/c each iteration takes less energy due to slack fill

DVS Architectures

Single Function Single Rate



Single V_{DD}



Multi-V_{DD} Architecture

10

Since optimal energy > proc. rate == work, next logical step is to have multiple processing rates

DVS Architectures

Single Function Single Rate

Add Temporal Granularity

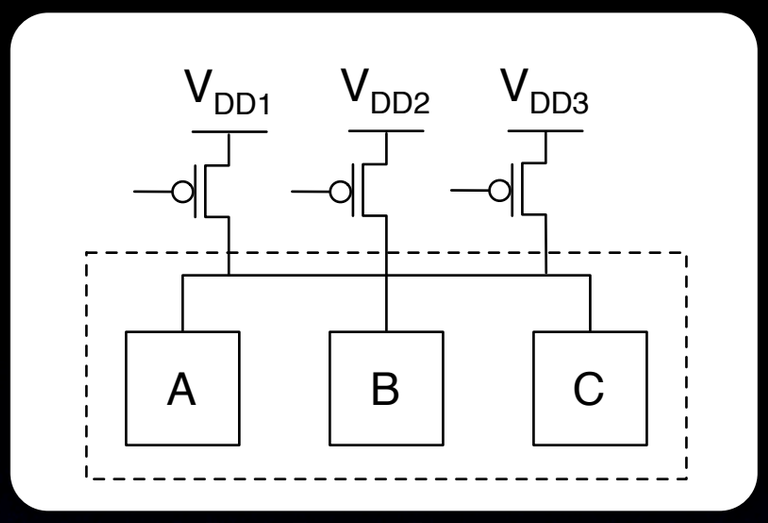
(Rate switching)

Single V_{DD}

Multi-V_{DD} Architecture

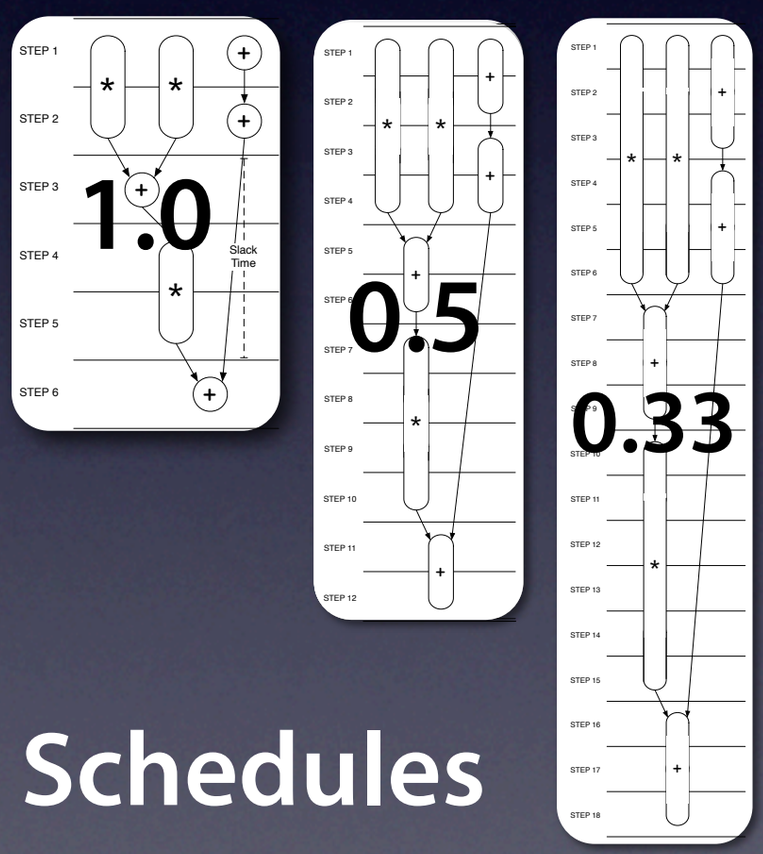
10

Since optimal energy > proc. rate == work, next logical step is to have multiple processing rates



Global DVS Architecture

Workload: 0.4
Processing Rate: 0.5

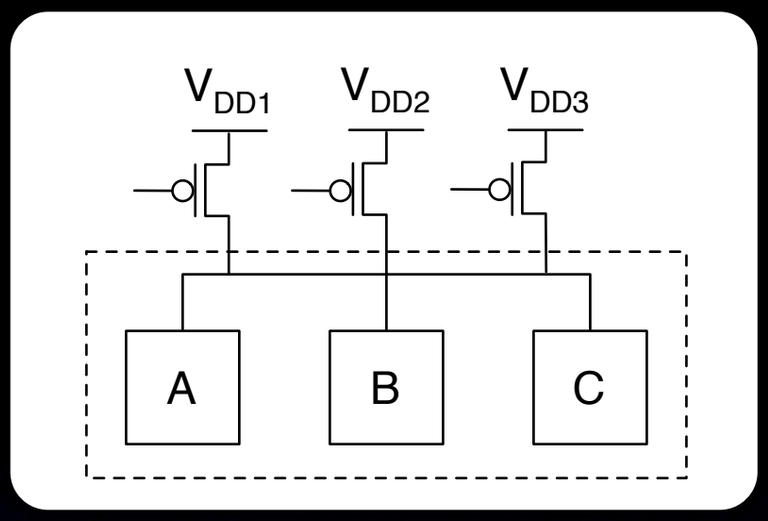


Schedules of Execution

1 now have three rates to choose from, can choose rate closer to workload

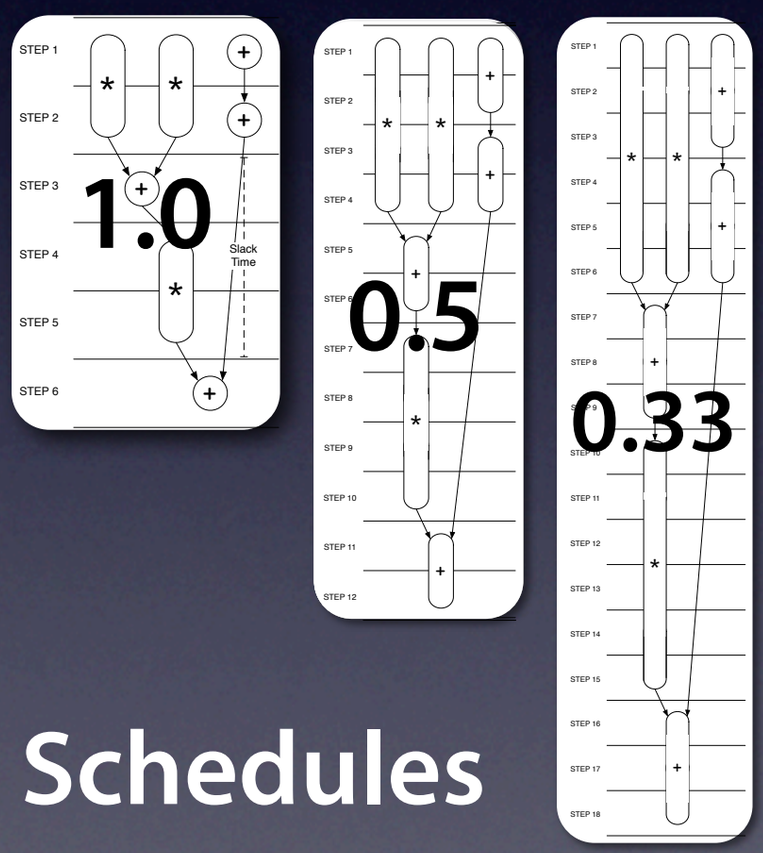
2 schedule still has slack due to coarse spatial gran.

3 multi-VDD operates in the same way but with sub-block energy savings



Global DVS Architecture

Workload: 0.4
Processing Rate: 0.5

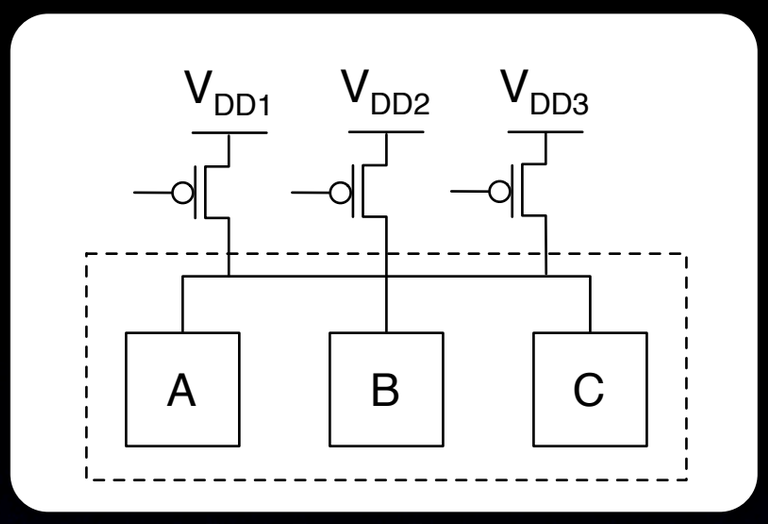


Schedules of Execution

1 now have three rates to choose from, can choose rate closer to workload

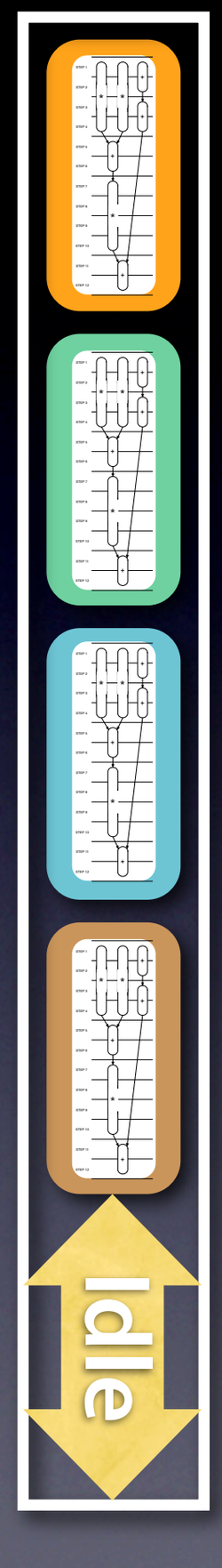
2 schedule still has slack due to coarse spatial gran.

3 multi-VDD operates in the same way but with sub-block energy savings

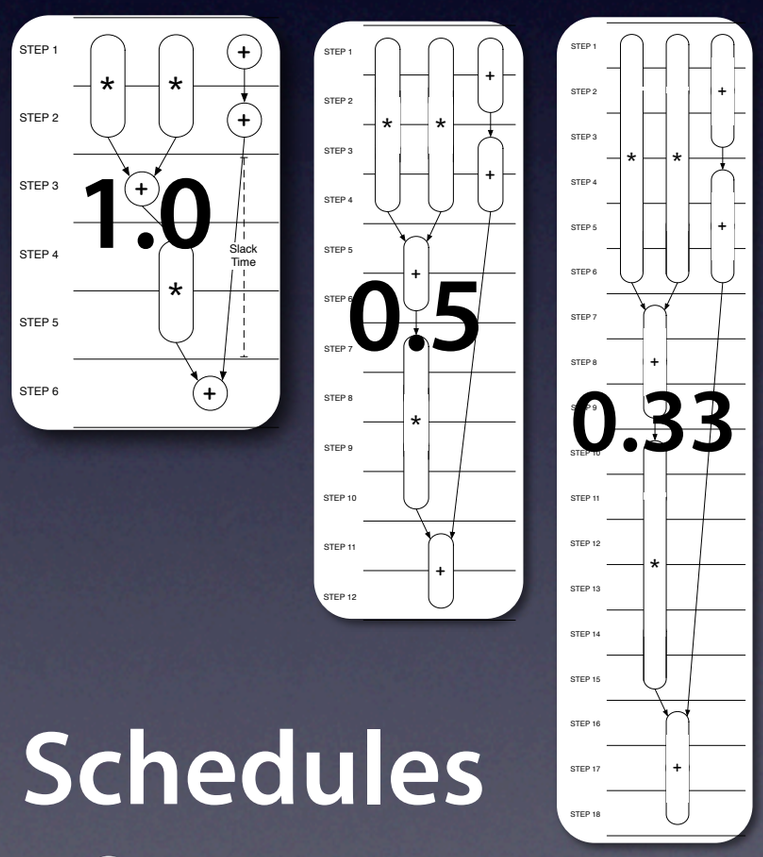


Global DVS Architecture

Workload: 0.4
Processing Rate: 0.5



Workload executed at lower rate



Schedules of Execution

Less slack time

1 now have three rates to choose from, can choose rate closer to workload

2 schedule still has slack due to coarse spatial gran.

3 multi-VDD operates in the same way but with sub-block energy savings

12

Novelty of approach: combination of fine grained switches + low switch overhead = many processing rates + less energy per iteration 2 Spatial gran. allows control over individual operations 3 Temporal gran. enables fast switching among static schedules

Fine Spatial Granularity (Sub-block voltage control)

12

Novelty of approach: combination of fine grained switches + low switch overhead = many processing rates + less energy per iteration 2 Spatial gran. allows control over individual operations 3 Temporal gran. enables fast switching among static schedules

Fine Spatial Granularity
(Sub-block voltage control)

+

Fine Temporal Granularity
(Voltage dithering)

12

Novelty of approach: combination of fine grained switches + low switch overhead = many processing rates + less energy per iteration 2 Spatial gran. allows control over individual operations 3 Temporal gran. enables fast switching among static schedules

Fine Spatial Granularity
(Sub-block voltage control)

+

Fine Temporal Granularity
(Voltage dithering)

=

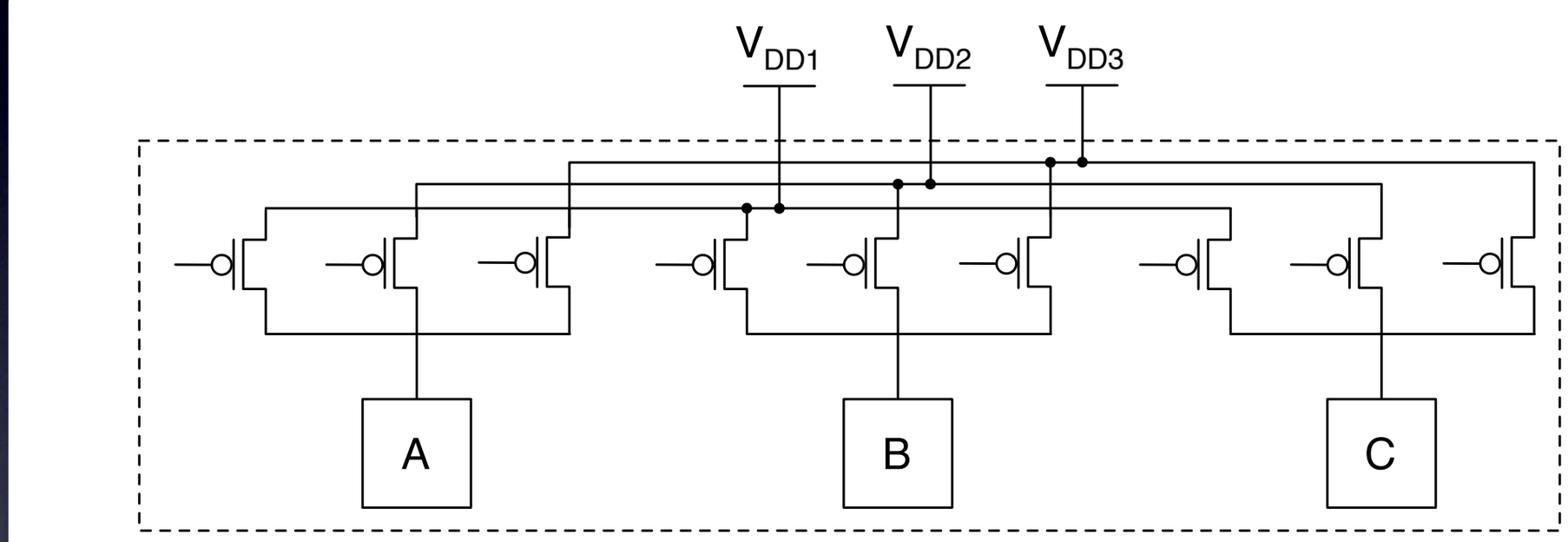
PDVS

12

Novelty of approach: combination of fine grained switches + low switch overhead = many processing rates + less energy per iteration 2 Spatial gran. allows control over individual operations 3 Temporal gran. enables fast switching among static schedules

DVS Architectures

Single-Function Multi-Rate

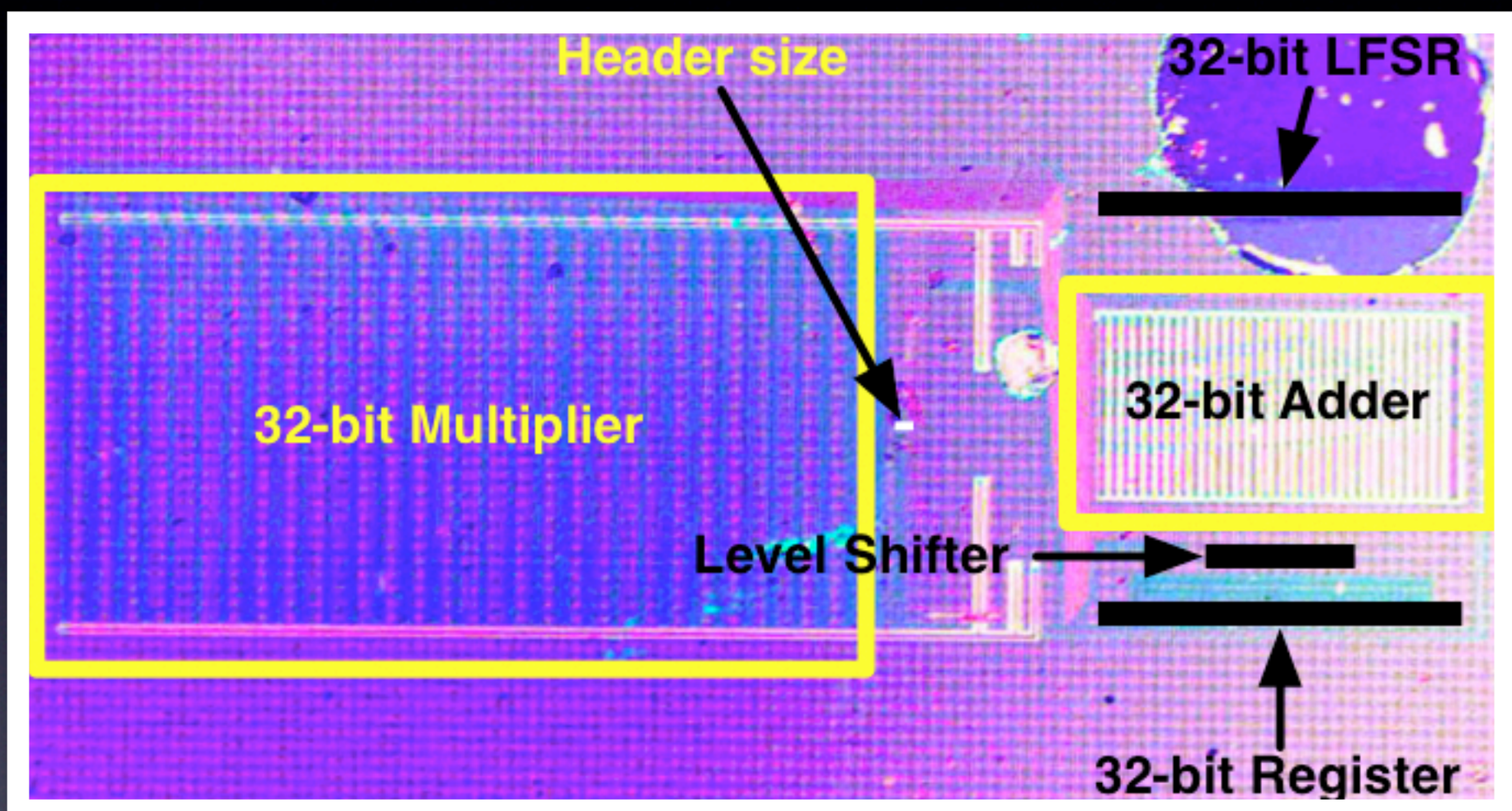


PDVS Architecture

13

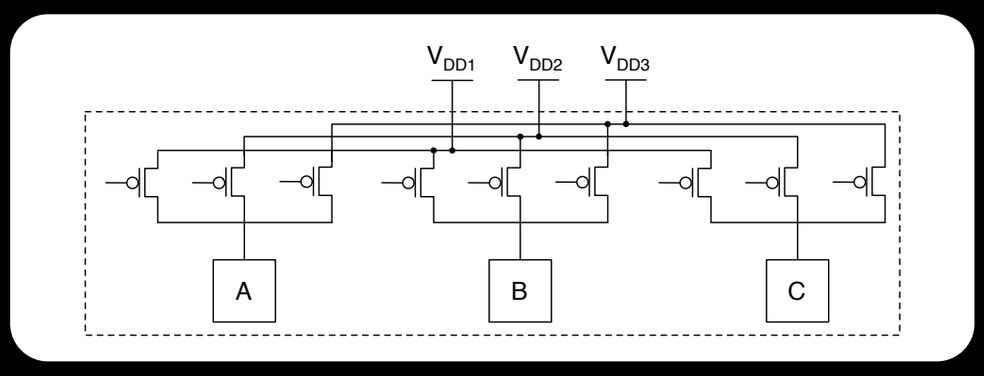
Enables sub-block voltage control + quickly switch between processing rates, dc-dc conv. for global dith.

90nm Test Chip

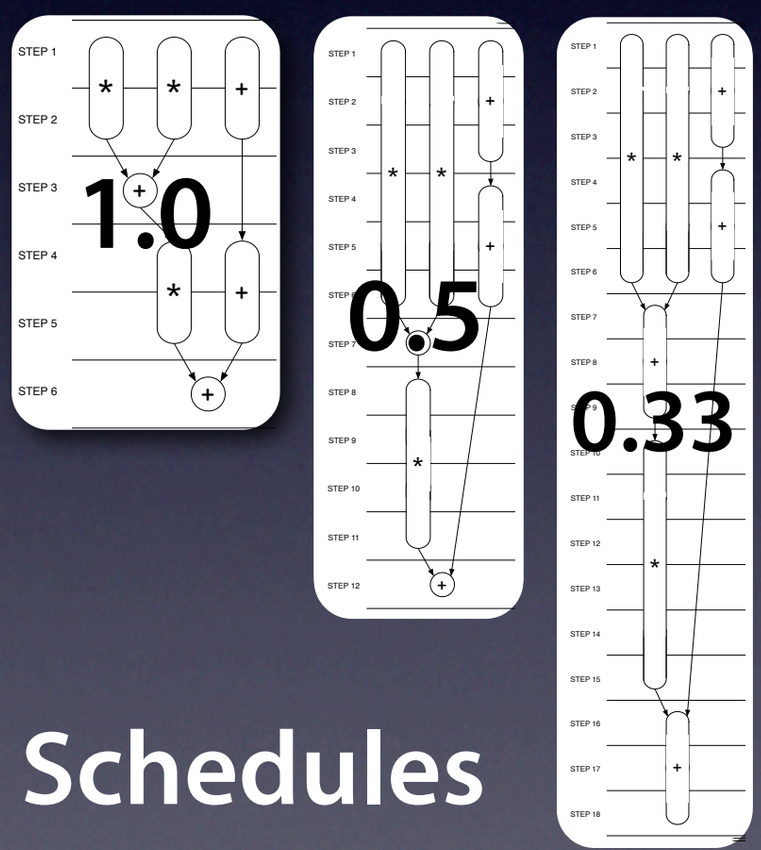


1 small overhead introduced by header switches and level shifter

2 global VDD rails were sized sufficiently large so sizing of header switches dominates delay, energy overhead



PDVS Architecture

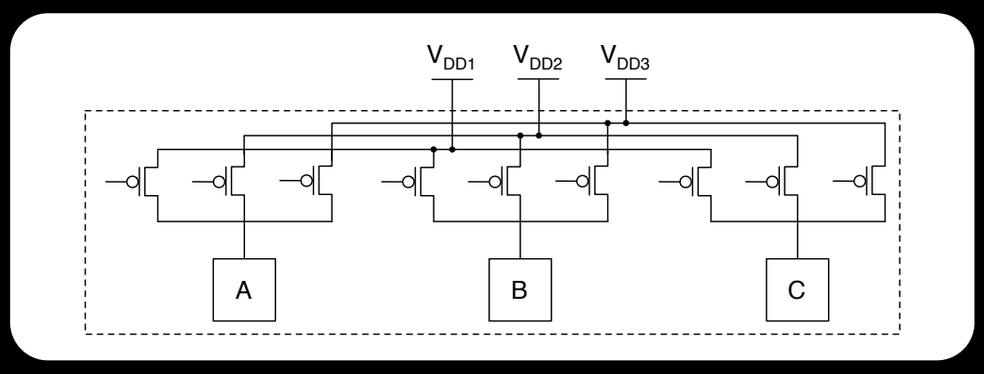


Schedules of Execution

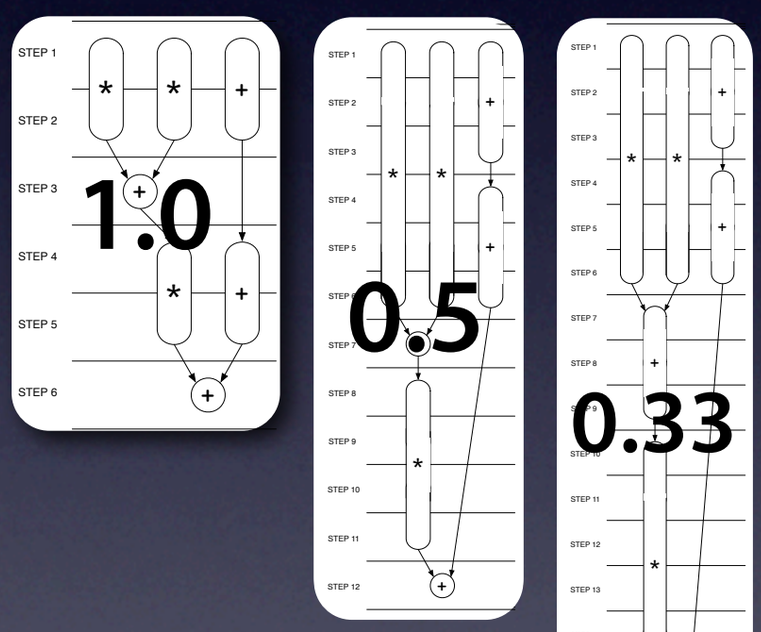
Workload: 0.4
Processing Rate: 0.415

15

Can achieve processing rate dramatically closer to workload > dithering, sub-block savings, less area than multi-VDD

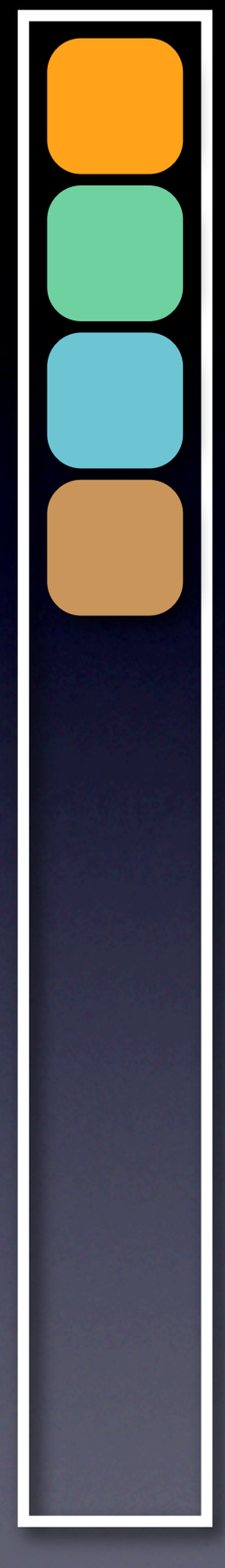


PDVS Architecture



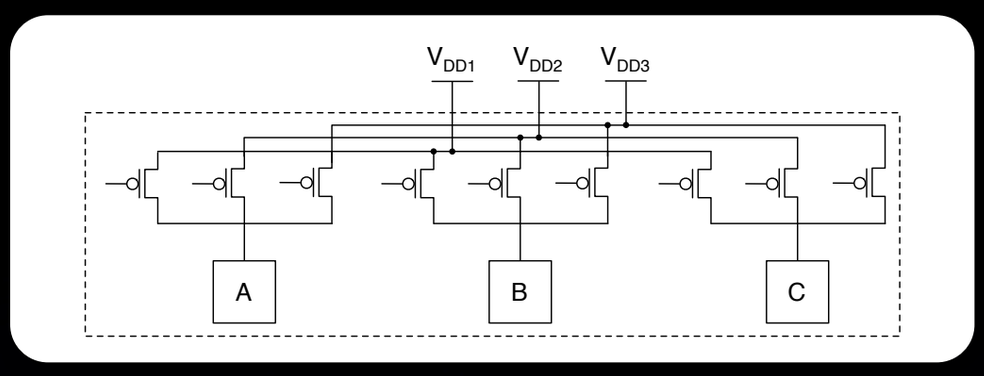
Schedules of Execution

Workload: 0.4
Processing Rate: 0.415

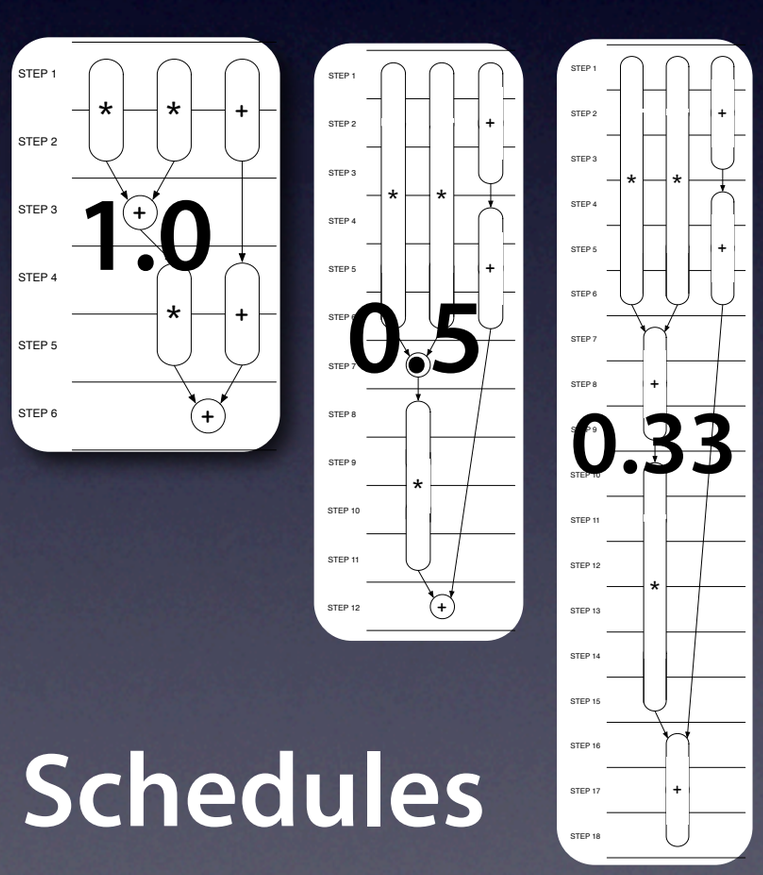


15

Can achieve processing rate dramatically closer to workload > dithering, sub-block savings, less area than multi-VDD



PDVS Architecture

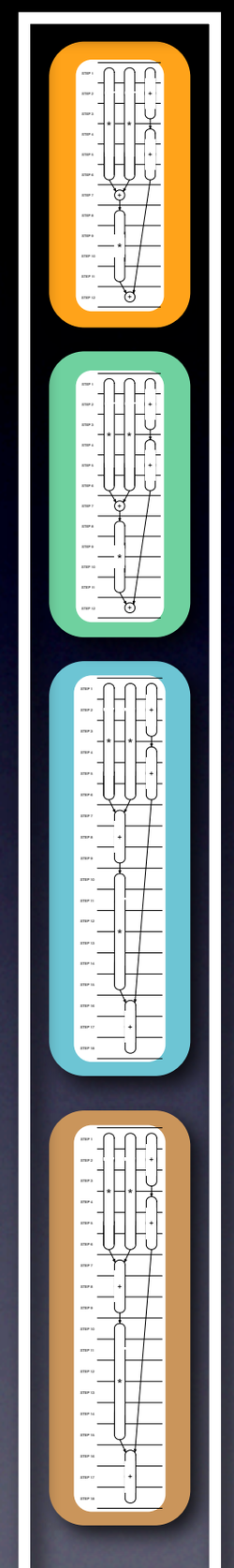


Schedules of Execution

Workload: 0.4
Processing Rate: 0.415

Processing rate approximates workload

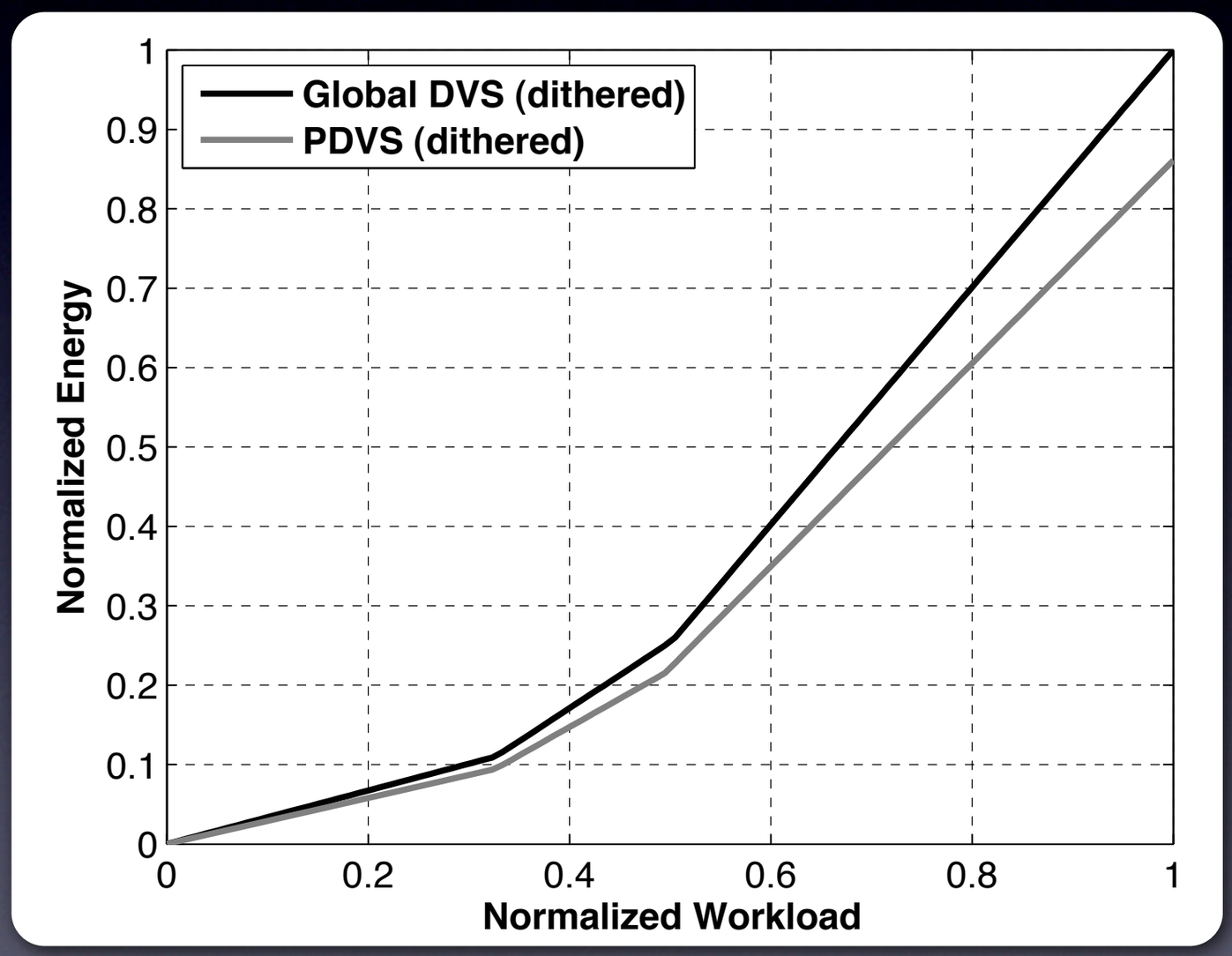
Dithering and **sub-block energy savings** implemented with **less area** than Multi-V_{DD}



15

Can achieve processing rate dramatically closer to workload > dithering, sub-block savings, less area than multi-VDD

DVS Architectures



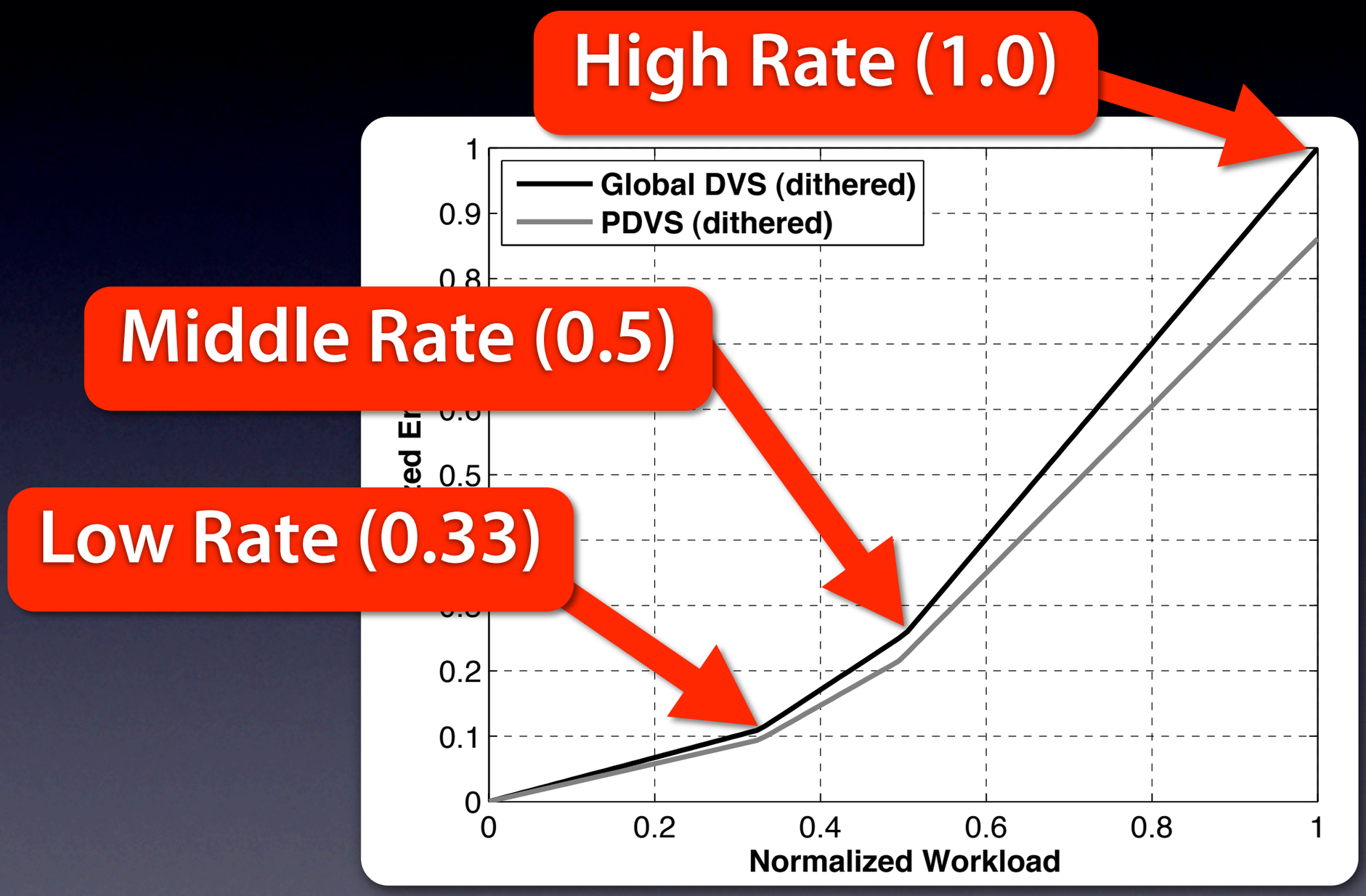
16

1 max. rate is lower due to sub-block energy savings, enabled by spatial gran.

2 Elbows occur at the three static processing rates

3 EXPLAIN IDEAL > very close

DVS Architectures



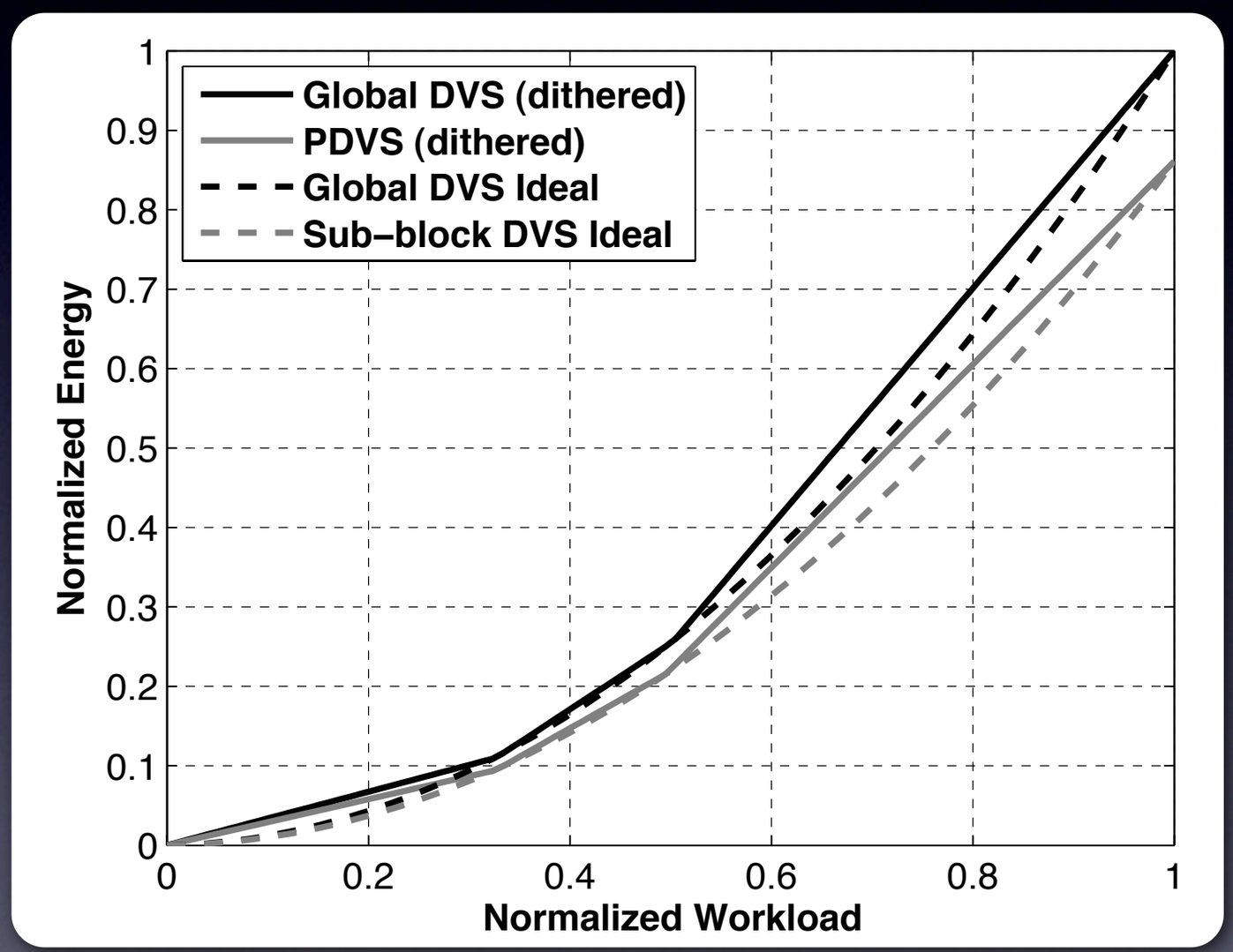
16

1 max. rate is lower due to sub-block energy savings, enabled by spatial gran.

2 Elbows occur at the three static processing rates

3 EXPLAIN IDEAL > very close

DVS Architectures



16

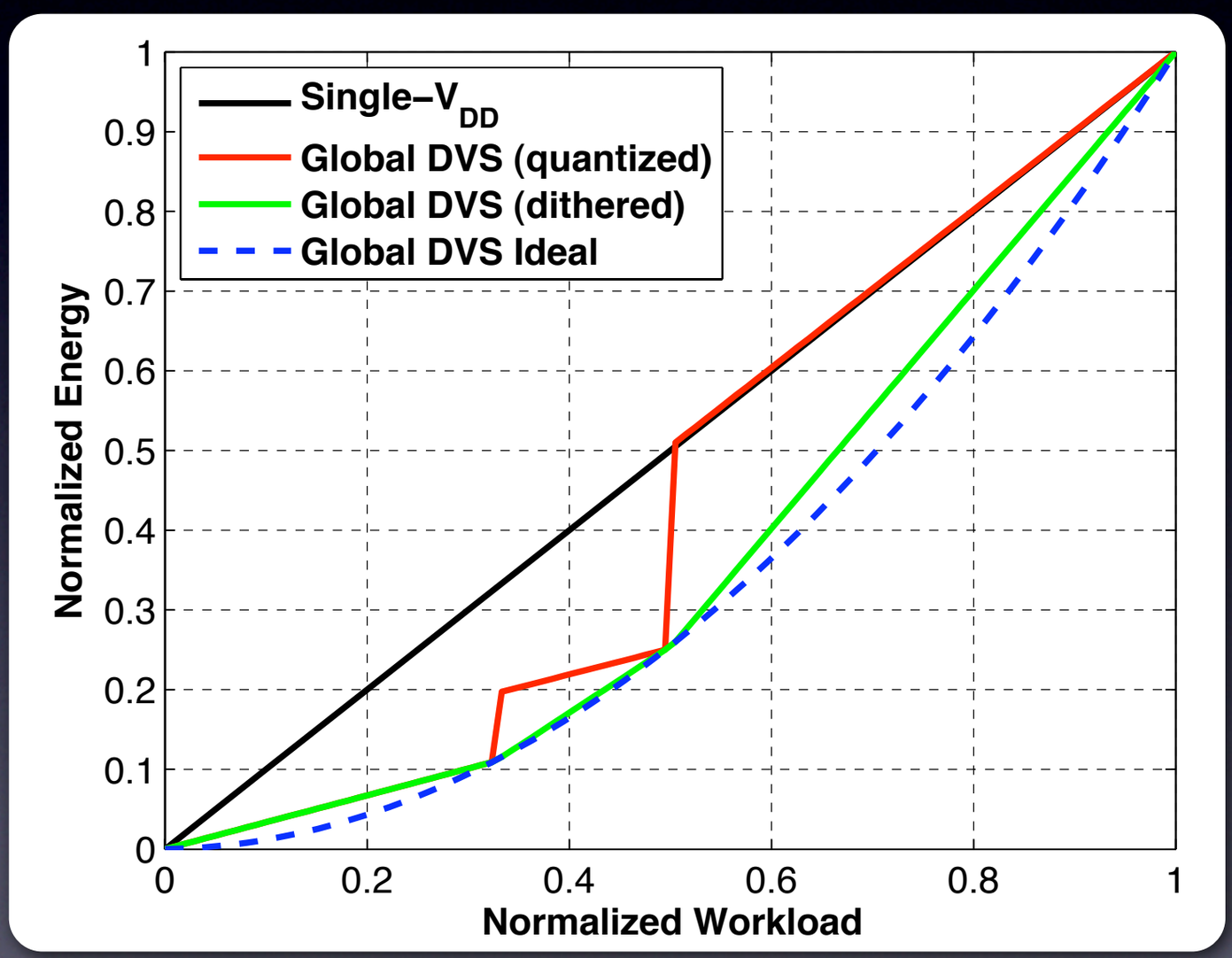
1 max. rate is lower due to sub-block energy savings, enabled by spatial gran.

2 Elbows occur at the three static processing rates

3 EXPLAIN IDEAL > very close

DVS Architectures

Global DVS Comprehensive Plot

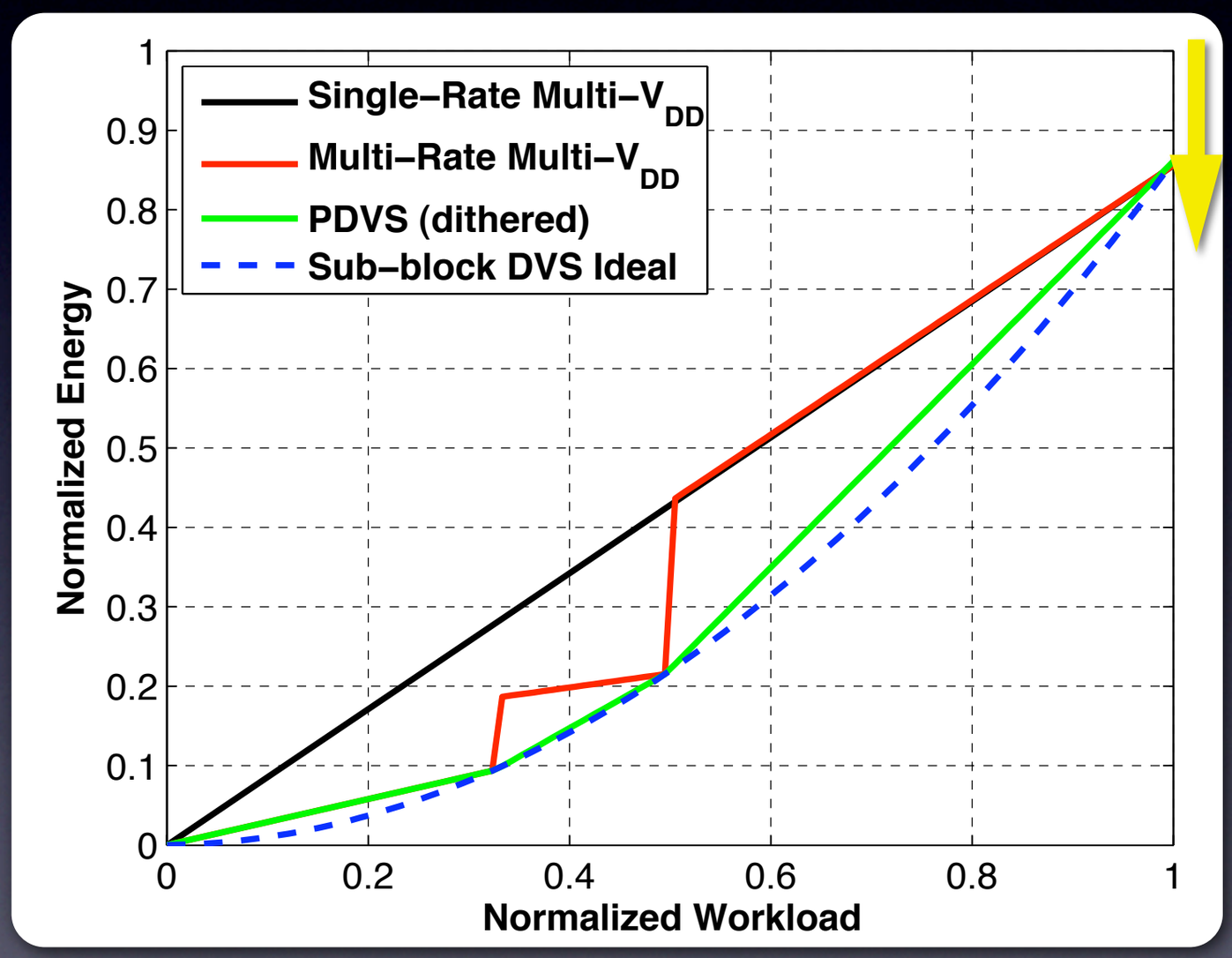


17

1. Global block architectures 2. sub-block architectures

DVS Architectures

Sub-Block DVS Comprehensive Plot



17

1. Global block architectures 2. sub-block architectures

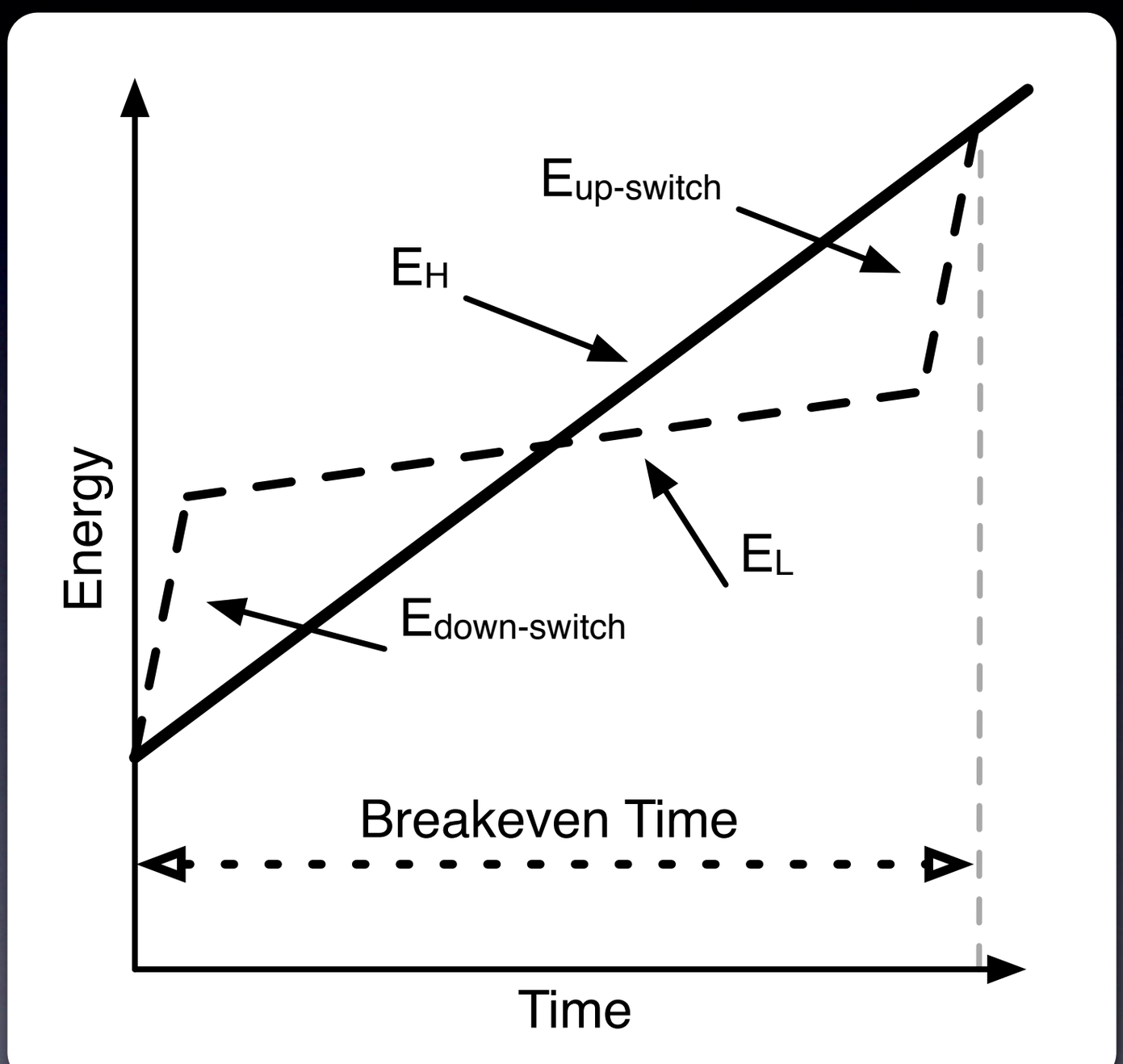
PDVS Overheads

- Delay: How long does a switch take?
- Energy: How much energy does it take?
- Measured overheads from test chip implemented in 90nm process

18

Next logical step is to ask, what are the overheads of PDVS: IDEAL SCHEDULES unrealistic due to real ovh

Breakeven Time



19

To find the point at which switching down to a lower proc. rate vs. staying at the high rate consumes less energy

Summary of Measurements

Multiplier

V_{DD} (V)	1.0	0.77	0.67
Delay (ns)	2.96	4.31	6.50
Active E per Op (pJ)	61.2845	35.5879	28.3758
Add'l Leak per Op (pJ)	0.0013	0.0107	0.0055
Total E per Op (pJ)	61.2845	35.5879	28.3758
Normalized Delay (to Adder)	6.9613	10.1274	15.2947
Nearest Delay (Adder cycles)	7	11	16
Switching Overhead (pJ)	-	6.7085	9.6715
Breakeven Cycles	-	1.2156	1.3853

20

Notice: 1. Inverse-quadratic trend in energy per op, cmp. to adder 2. Nonlin. trend delay 3. Cmp. magnit. of E per op to sw. Ovh. 4. brkevn cycls < 1 op

Summary of Measurements

Multiplier

V_{DD} (V)	1.0	0.77	0.67
Delay (ns)	2.96	4.31	6.50
Active E per Op (pJ)	61.2845	35.5879	28.3758
Add'l Leak per Op (pJ)			
Total E per Op (pJ)	61.2845	35.5879	28.3758
Normalized Delay (to Adder)	6.9613	10.1274	15.2947
Nearest Delay (Adder cycles)	7	11	16
Switching Overhead (pJ)	-	6.7085	9.6715
Breakeven Cycles	-	1.2156	1.3853

Quadratic Energy Trend

20

Notice: 1. Inverse-quadratic trend in energy per op, cmp. to adder 2. Nonlin. trend delay 3. Cmp. magnit. of E per op to sw. Ovh. 4. brkevn cycls < 1 op

Summary of Measurements

Multiplier

V_{DD} (V)	1.0	0.77	0.67
Delay (ns)	2.96	4.31	6.50
Active E per Op (pJ)	61.2845	35.5879	28.3758
Add'l Leak per Op (pJ)	0.0013	0.0107	0.0055
Total E per Op (pJ)	61.2845	35.5879	28.3758
Normalized Delay (to Adder)			
Nearest Delay (Adder cycles)	7	11	16
Switching Overhead (pJ)	-	6.7085	9.6715
Breakeven Cycles	-	1.2156	1.3853

Nonlinear Scaling Trend

20

Notice: 1. Inverse-quadratic trend in energy per op, cmp. to adder 2. Nonlin. trend delay 3. Cmp. magnit. of E per op to sw. Ovh. 4. brkevn cycls < 1 op

Summary of Measurements

Multiplier

V_{DD} (V)	1.0	0.77	0.67
Delay (ns)	2.96	4.31	6.50
Active E per Op (pJ)	61.2845	35.5879	28.3758
Add'l Leak per Op (pJ)	0.0013	0.0107	0.0055
Total E per Op (pJ)	61.2845	35.5879	28.3758
Normalized Delay (to Adder)	6.9613	10.1274	15.2947
Nearest Delay (Adder cycles)	7	11	16
Switching Overhead (pJ)	-	6.7085	9.6715
Breakeven Cycles	-	1.2156	1.3853

20

Notice: 1. Inverse-quadratic trend in energy per op, cmp. to adder 2. Nonlin. trend delay 3. Cmp. magnit. of E per op to sw. Ovh. 4. brkevn cycls < 1 op

Summary of Measurements

Multiplier

V_{DD} (V)	1.0	0.77	0.67
Delay (ns)	2.96	4.31	6.50
Active E per Op (pJ)	61.2845	35.5879	28.3758
Add'l Leak per Op (pJ)	0.0013	0.0107	0.0055
Total E per Op (pJ)	61.2845	35.5879	28.3758
Normalized Delay (to Adder)	6.9613	10.1274	15.2947
Nearest Delay (Adder cycles)	7	11	16
Switching Overhead (pJ)	-	6.7085	9.6715
Breakeven Cycles	-	1.2156	1.3853

20

Notice: 1. Inverse-quadratic trend in energy per op, cmp. to adder 2. Nonlin. trend delay 3. Cmp. magnit. of E per op to sw. Ovh. 4. brkevn cycls < 1 op

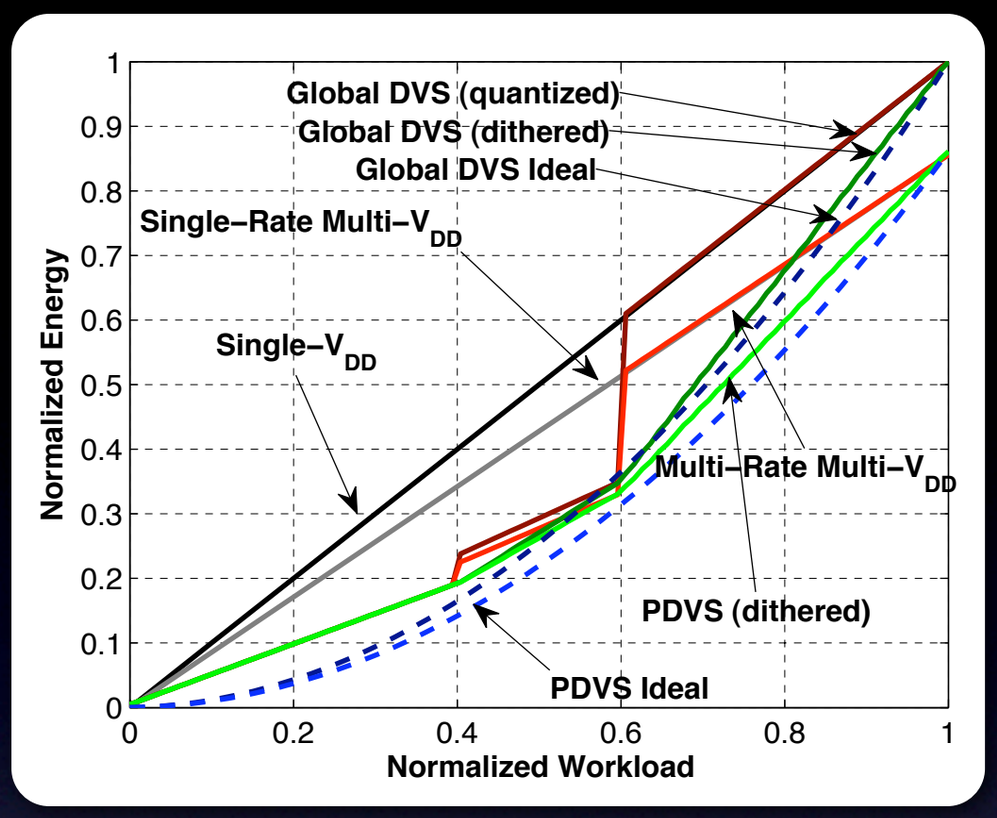
Observations

Single-Function Single-Rate

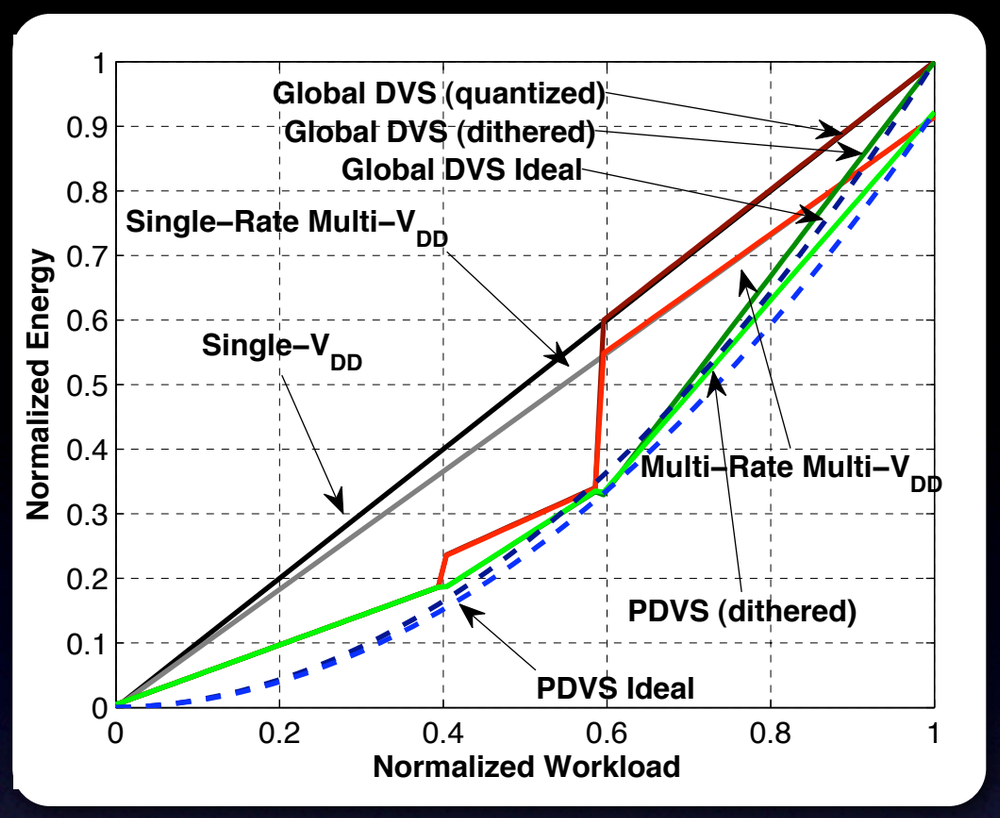
- vs. Single-V_{DD}:
  - Energy ↓ due to fine spatial granularity (sub-block savings)
  - Area ↕ depending on schedule
- vs. Multi-V_{DD}:
  - Energy ↑ due to switching overhead
  - Area ↓ due to multi-modality

21

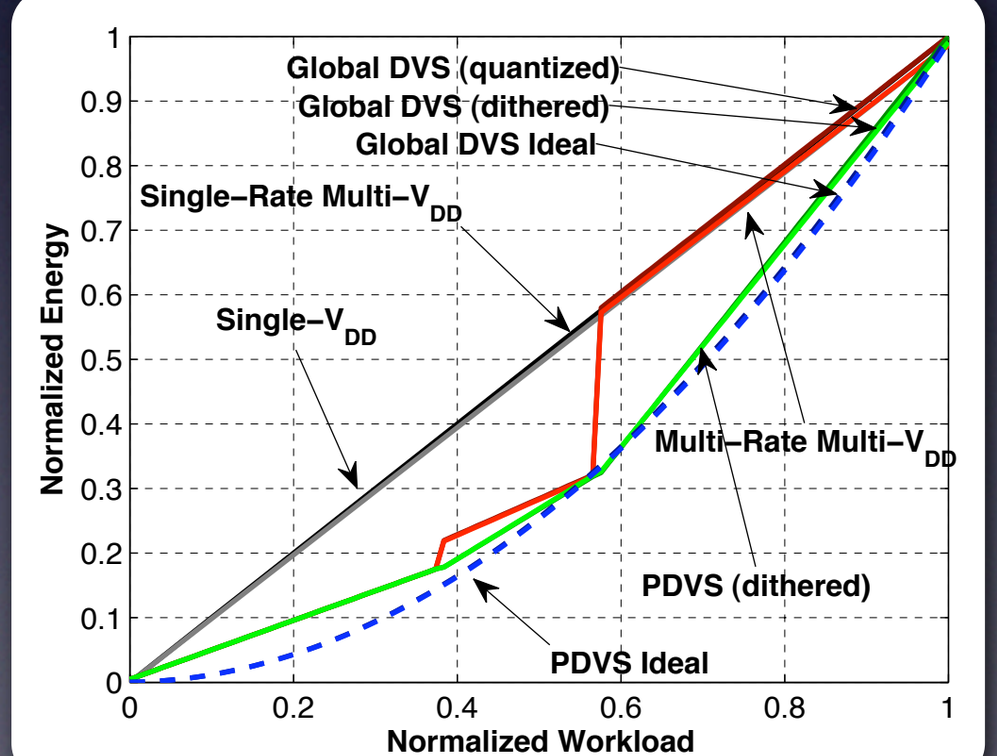
Observations for Single-Function Single-Rate If we hold schedule constant



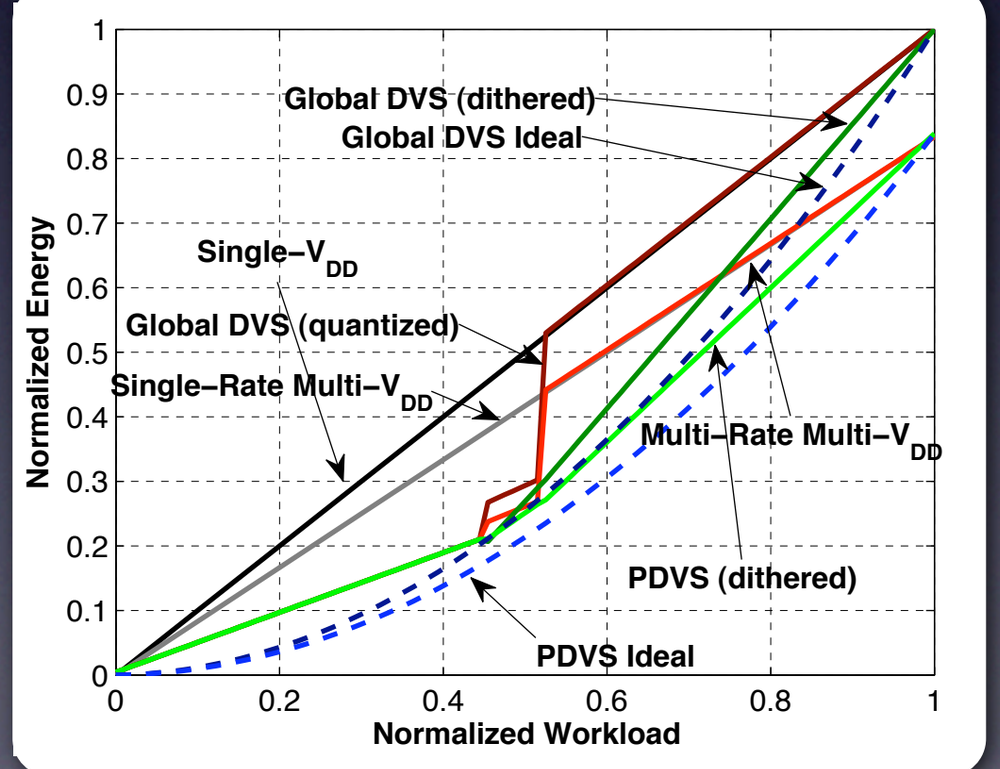
AR Lattice



DiffEQ



Ellip

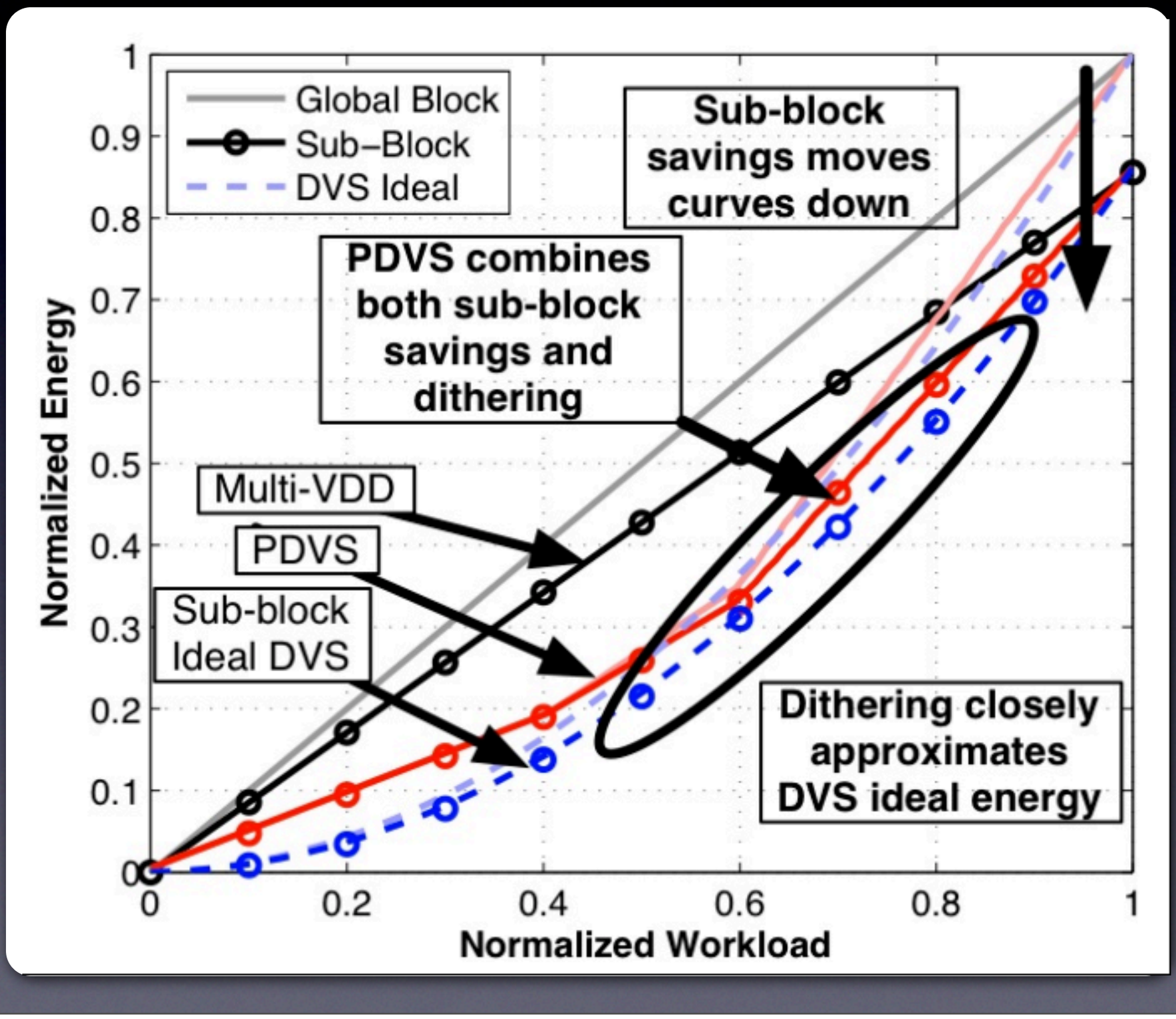


FIR

22

1. slack time of schedules 2. elbows of quantized architectures do not occur in the same places

SFMR Results



Observations

Single-Function Multi-Rate

- vs. Global DVS:
  - Energy ↓ due to fine spatial granularity (sub-block savings)
  - Area ↑ depending on schedule
- vs. Multi-Rate Multi-V_{DD}:
  - Energy ↓ due to fine temporal granularity (voltage dithering)
  - Area ↓ due to multi-modality

24

Observations for Single-Function Multi-Rate

Results

Multi-Function Multi-Rate

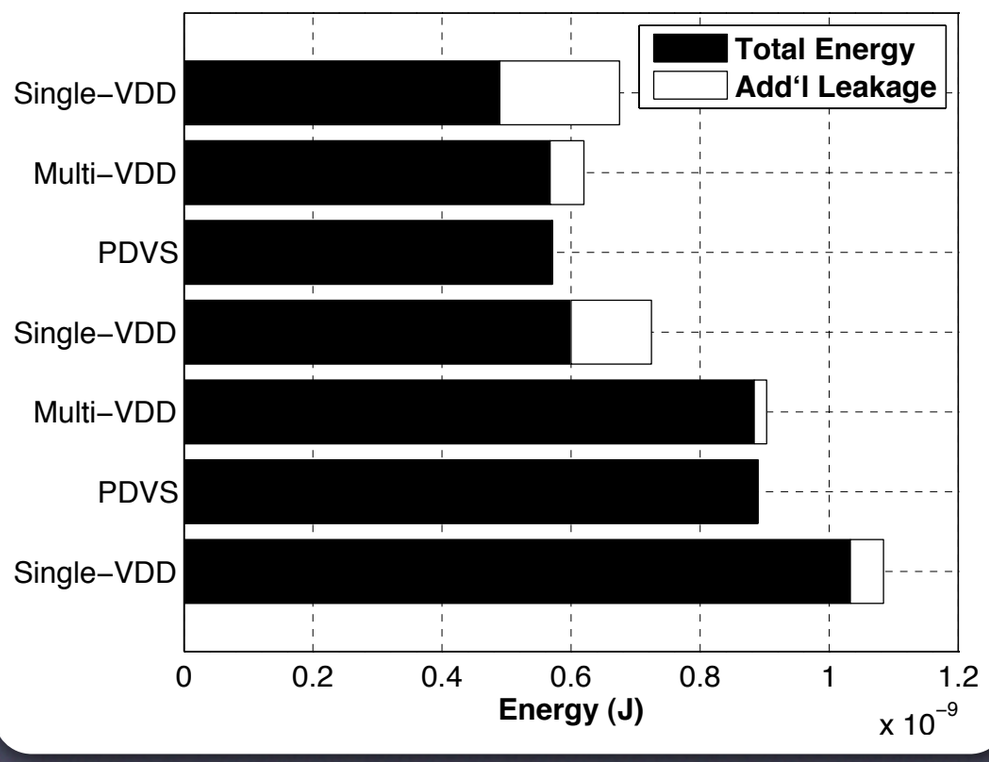


Table 5.6: Component counts in multi-mode

	PDVS	Multi-V_{DD}	Single-V_{DD}
VH Add	0	4	4
VM Add	0	4	4
VL Add	0	3	4
Total Add	7	11	12
VH Mult	0	5	8
VM Mult	0	4	8
VL Mult	0	4	8
Total Mult	8	13	24

AR Lattice

25

Architectures have enough components to implement all benchmarks 2 PDVS saves dynamic energy w/efficient schedules, 3 static energy w/fewer components > high utilization

Observations

Multi-Function Multi-Rate

- vs. Global DVS:
  - Energy ↓ due to fine spatial, temporal granularity, reduction in leakage
  - Area ↑ due to parallelism
- vs. Multi-Rate Multi-V_{DD}:
  - Energy ↓ due to fine temporal granularity, reduction in leakage
  - Area ↓ due to multi-modality

26

Observations for Multi-Function Multi-Rate

Conclusions

- PDVS combines fine spatial, temporal granularity with sub-block headers
- Test chip fabricated
- Breakeven time < 1 operation
- Switching delay < 1 clock
- Area-efficient energy savings

header sizing rule of thumb

what are the real implications, through PDVS that you get closest to the ideal energy curve, dithering + sub-block savings is the biggest implication, in an area efficient way

References

1. Putic, Mateja, Liang Di, Benton H. Calhoun, John Lach. “Panoptic DVS: A Fine-Grained Dynamic Voltage Scaling Framework for Energy Scalable CMOS Design”. ICCD2009, October 2009.

2. Liang Di, Mateja Putic, John Lach, and Benton H. Calhoun."Power Switch Characterization for Fine-Grained Dynamic Voltage Scaling" ICCD2008, pages 605-611, October 2008.

3. “Investigating Fine-Grained Dynamic Voltage Scaling for Low Power CMOS” Master’s Thesis, University of Virginia, 2008.

28