A Dynamic-Programming Algorithm for Reducing the Energy Consumption of Pipelined System-Level Streaming Applications

N. Liveris, H. Zhou, P. Banerjee*
Northwestern University, Evanston IL, USA
*HP Labs, Palo Alto Ca, USA

ASP-DAC 2008, Seoul, Korea
January 22, 2007

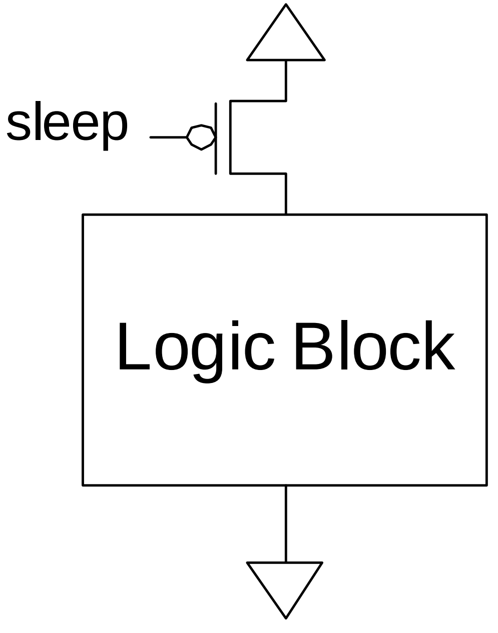
Outline

- Motivation
- Technique
- Theoretical Exploration
- Dynamic Programming Solution
- Experimental Results
- Conclusions

Motivation

- Leakage power consumption is expected to become dominant for future technologies
- Techniques for reducing power consumption are needed that
  - are adaptive to environment changes
  - do not require resynthesis of IP cores

Power Gating

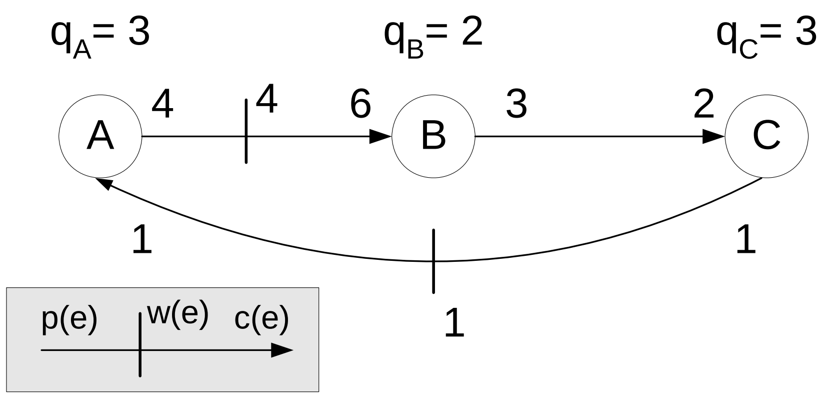


- Sleep transistor creates "virtual Vdd" when logic block is idle (sleep = 1)
- Stand-by energy consumption is decreasing
  - $E = T\, I_L(Vdd)\, Vdd$

Power Gating

- With power gating a module is shut down when it is idle
- Energy penalty mainly for switching from sleep to active – loading of the nodes back to normal Vdd levels
- In this work we try to increase the energy savings by reducing the number of switches

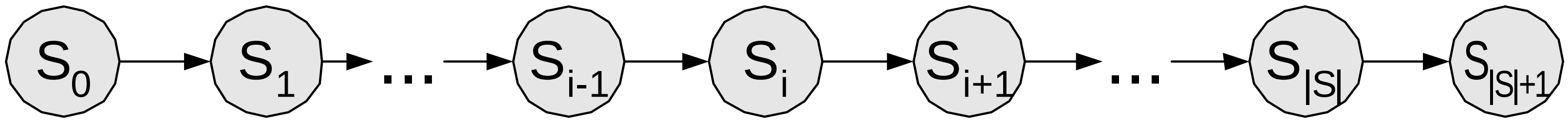
Synchronous Dataflow Graphs



If for all edges p(e)=c(e)=1, the graph is called unirate.

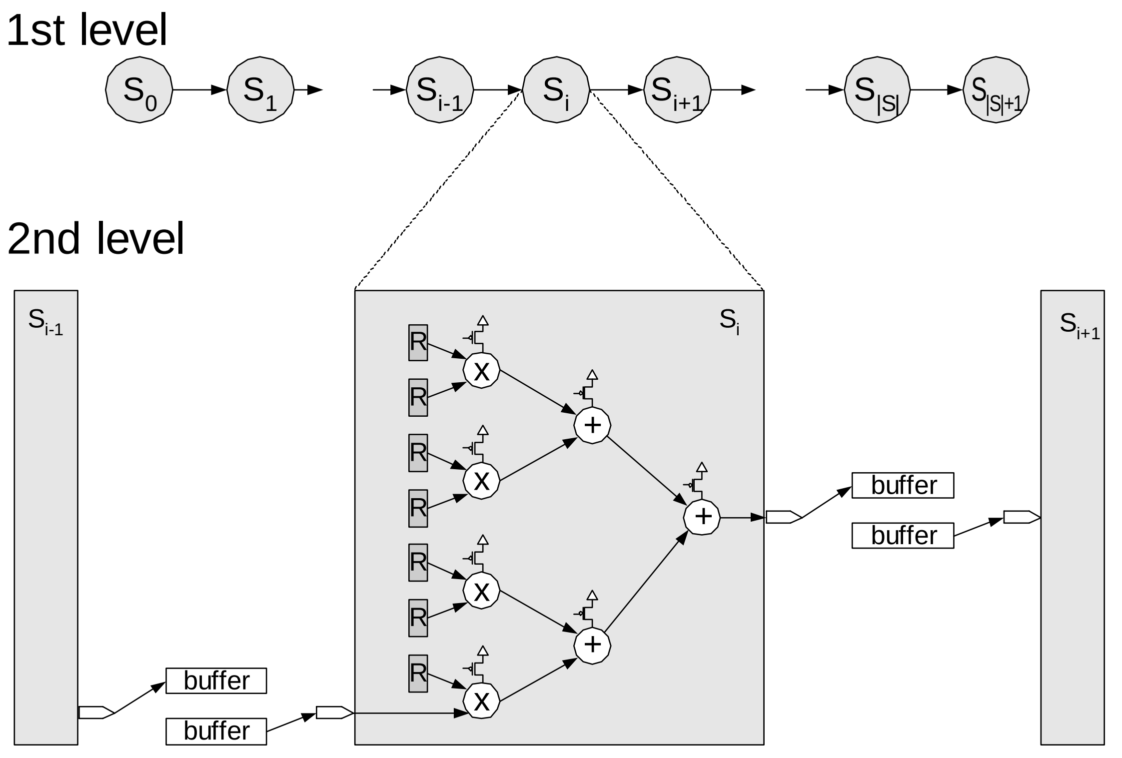
- Each node represents computation process
  - constant production and consumption rate
  - executed a specific number of times during each complete cycle
- Edge represents a channel between two actors
  - FIFO protocol for tokens
  - initial number of tokens on edge (delays)

Chain-Structured Synchronous Data Flow Graphs



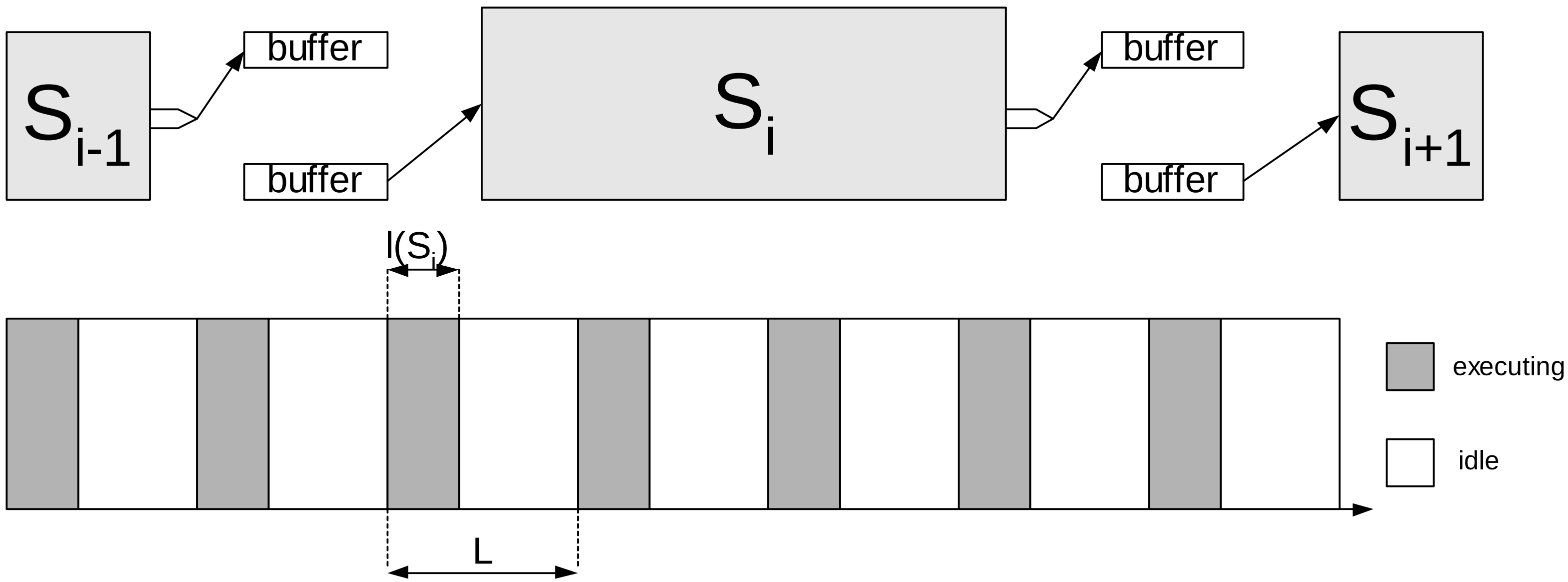
- First level of hierarchy, chain-structured Synchronous Data Flow Graph
- Each node S_i reads data produced by S_{i-1} and produces data for S_{i+1}
- Each node represents a pipeline stage and can be executed in parallel with other nodes
- Model commonly used for pipelined streaming applications

System Description



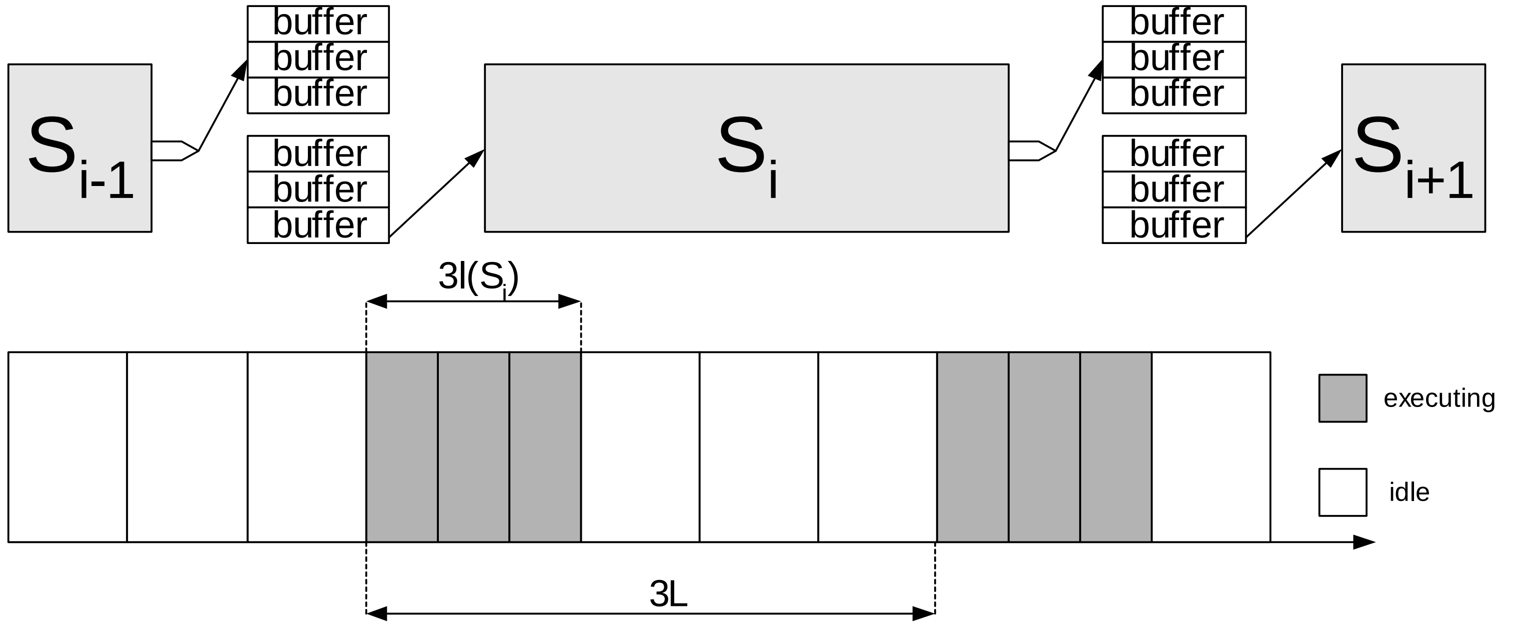
- Second level: Each stage is a graph
- Nodes of the graph represent hardware units (processes) that can be independently power gated
- The edges of the graph represent data flow

Stage Execution



- The interval *L-l(Si)* is the slack time for the stage Si
- If for a process v of stage Si power can be saved, the process is put to sleep mode while being idle
- The number of consecutive executions of each stage is 1 (x=1)

Changing the Stage Execution

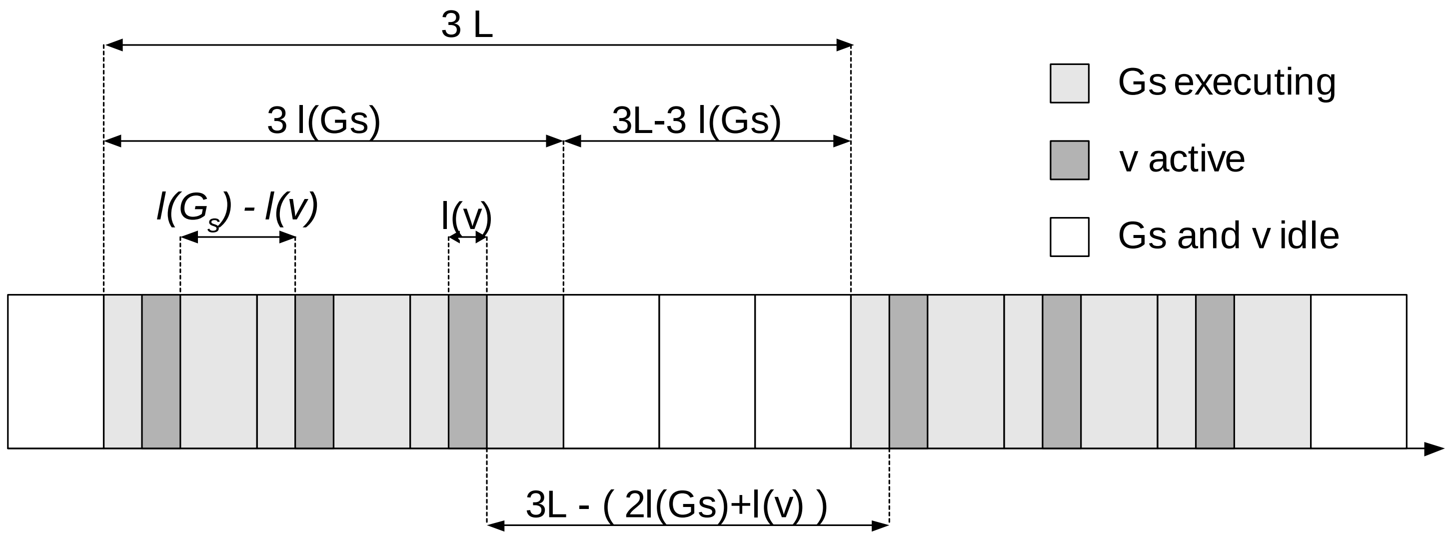


- The interval *3L-3l(Si)* is the slack time for the stage Si
- More processes of stage Si can be put in sleep mode due to the increased idle time
- For some processes the penalty of switching mode is paid only once in 3L
- The number of consecutive executions of stage Si is 3 (x=3)

Problem Formulation

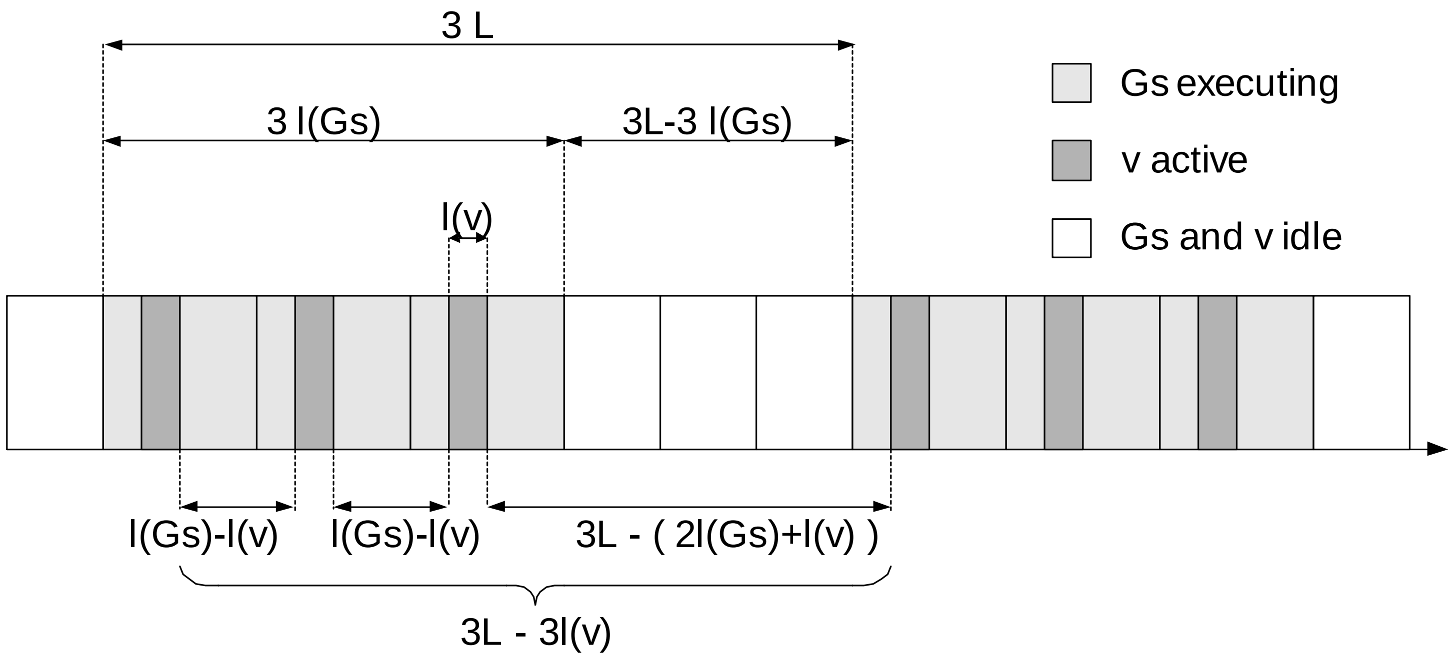
- Determine the number of consecutive executions for each pipeline stage, so that the energy savings will be maximized
  - keep the average throughput of the application constant
  - take into account the energy penalty caused by the increase in the number of buffers

Type-1 Processes



- If for process *v* of stage *Gs* the idle time *l(Gs)-l(v)* is not enough to put *v* in sleep mode, then v is a type-1 process
- For type-1 processes the energy savings are an increasing function of the number of consecutive executions (*x*) of the stage *Gs*
- An upper bound for the energy savings in *L* cycles using this technique is derived

Type-2 Processes



- If for process *v* of stage G_S the idle time $l(G_S)-l(v)$ is enough to put *v* in sleep mode, then v is a type-2 process
- For type-2 processes the energy savings in *L* cycles are independent of the number of consecutive executions of G_S

Energy Penalty on Channels

- The energy penalty of an edge is an increasing, non-linear function of the number of buffers
- Number of buffers is increasing with respect to
  - the x value of the tail and head node (case 1)
  - the lcm of the x values of tail and head node (case 2 – unirate case)
  - the lcm of the x and q values of the tail and head node (case 3 – multirate case)
- Table that returns the energy penalty based on the x values of the tail and head node of the channel is input to the algorithm

Bound on x – quality metric

- Given a quality metric p as input to the algorithm, a bound x_{max} can be derived for the x values from the parameters of the graph (case 1 and case 2)
- In solution space $[1, x_{max}]^{|S|}$ there is at least one solution for which $E_{sav} > (1-p)\ E_{max}$
- For example if input p=0.02, there will be solution with $E_{sav} > 0.98\ E_{max}$ in the solution space
- The bound x_{max} is derived from the effect of an increase in x can have on the energy savings of type-1 processes

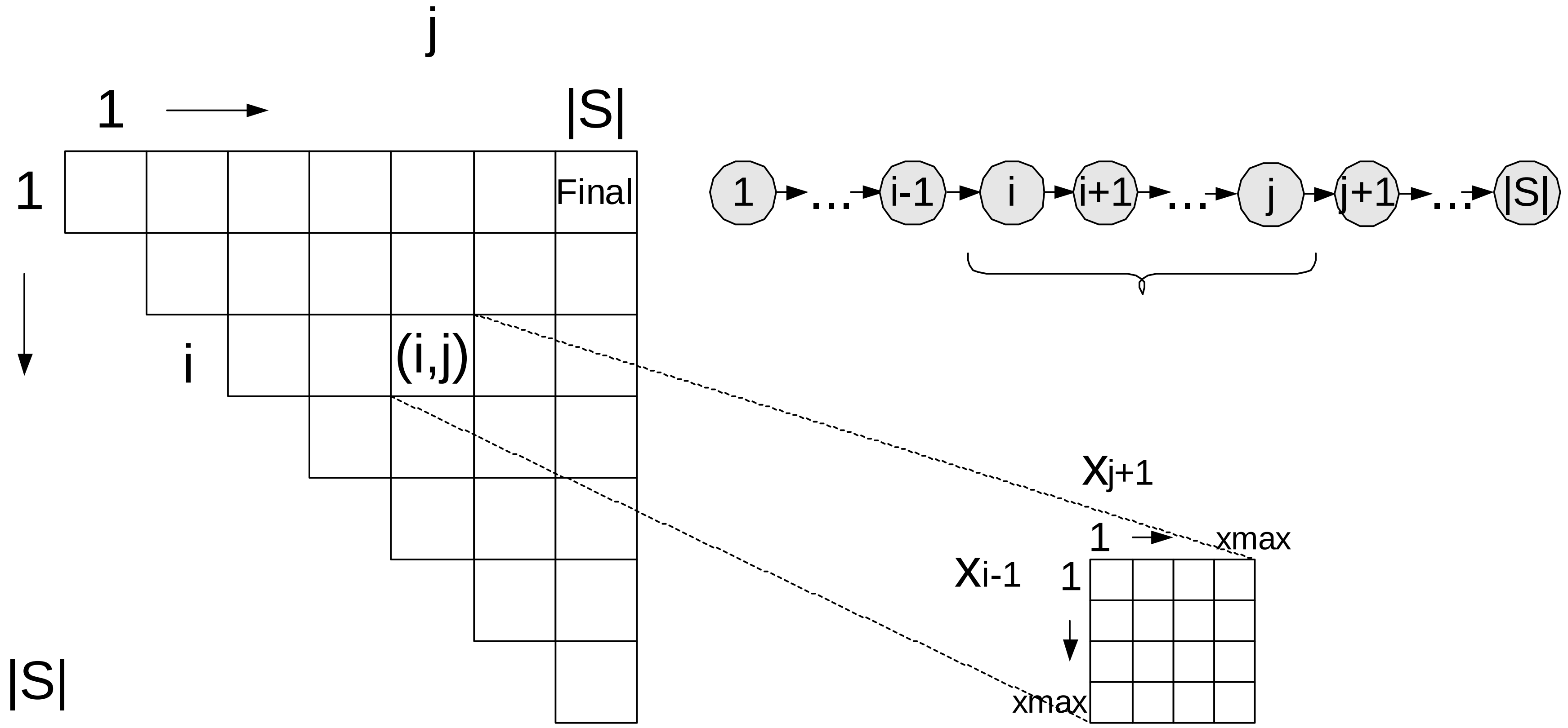
Bound on x – energy penalty

- Bound will be determined by energy penalty on channels between pipeline stages
- Theoretical limit on savings for type-1 processes
- Bound is derived when the penalty on a single edge exceeds all savings
- Bound is applicable to all three cases (increasing function, unirate graph, multirate graph)

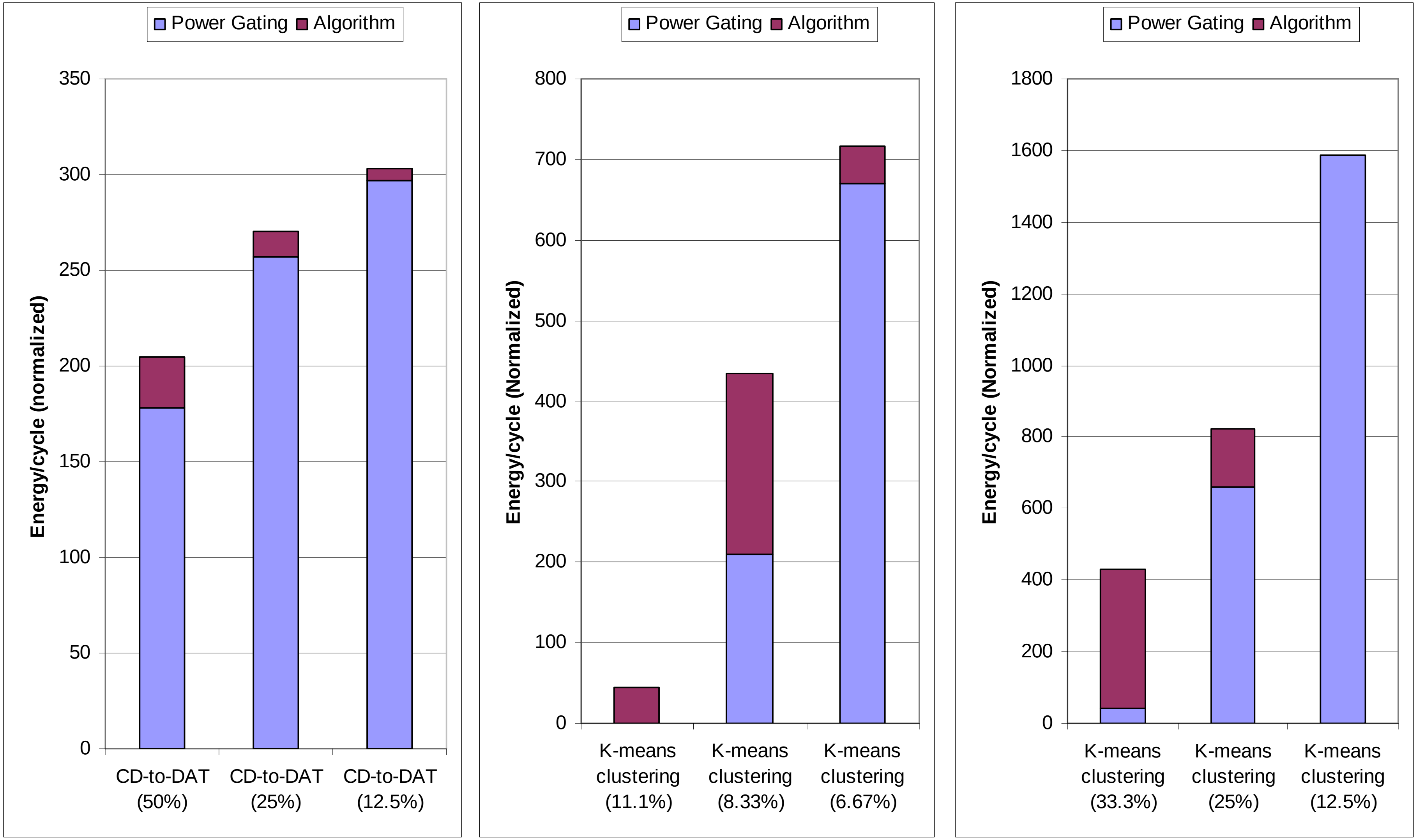
Dynamic Programming Algorithm - Intuition

- Even after determining x_{max}, the solution space is still large ($x_{max}^{|S|}$)
- For the x value of pipeline stage S_i, only the x values of stages S_{i-1} and S_{i+1} need to be taken into account
- Algorithm finds best solution for each stage, for any x values of the neighbor stages
- Then the algorithm combines the best solutions to solve the problem for each subchain, for any x values of the neighbor stages
- The solution returned by the algorithm maximizes the energy savings for the whole chain-structured graph under the restriction that for all stages x belongs to $[1, x_{max}]$

Dynamic Programming Algorithm



Experimental Results



Experimental Results

Application	Input Rate	Alg. Exec. Time (sec)	*xmax*	Increase in En. Savings
CD-to-DAT (multirate -3 stages)	50%	2.97	71	15.17%
	25%	2.97	71	5.25%
	12.50%	2.98	71	2.27%
K-means clustering (unirate - 10 stages)	11.10%	144.39	169	N/A
	8.33%	29.26	100	107.31%
	6.67%	3.37	49	6.71%
K-means clustering (unirate - 3 stages)	33.33%	5.79	100	900.38%
	25.00%	0.7	49	24.25%
	12.50%	0.06	16	0.00%

Application	10-stage K-means		3-stage K-means	
p	0.9	0.95	0.9	0.95
Input Rate	6.67%	6.67%	25.00%	25.00%
Alg. Exec. Time(sec)	3.37	351	0.7	44.2
xmax	49	225	49	196
Increase in En. Savings	6.71%	6.71%	24.25%	24.25%

Conclusions

- Presented an algorithm to increase energy savings of pipelined streaming applications
- Algorithm increases energy savings obtained from power gating by finding the number of consecutive executions of each pipeline stage
- Energy savings are larger when the slack is not enough for all hardware units to be put into sleep mode

Thank you!