Newcastle University

holistic energy harvesting

Power Adaptive Computing

Alex Yakovlev, School of EECE,
Newcastle University

Power Management Technologies Meeting, NPL
27 May 2010

Newcastle University

holistic energy harvesting

Outline

- Motivation
  - Energy proportional computing
  - Designing systems for harvested energy supplies
- Power-adaptive computing: design aspects
- Potential for asynchronous (self-timed) logic:
  - Robustness
  - Energy-efficiency
- Power adaptive research in Holistic project
  - Speed-independent SRAM
  - Power Sensor and Charge to Code Conversion
  - Run-time power modulation using dynamic scheduling
- Conclusion

Messages from ITRS

- Non-ideal device and supply/threshold voltage scaling leads to:
  - Leakage,
  - Power management and delivery
- We're entering the 2D world of progress: "More More" (scaling factor) and "More than Moore" (functional diversification) – so scaling is not everything to battle against!
- The "More than Moore" increasingly includes non-digital aspects – RF comms, power control, passive components, sensors, actuators etc.
- Design innovations (hardware and software) will help to reduce the design costs by 50-60 times and will have increasing impact in this 2D progress

Design For Low Power

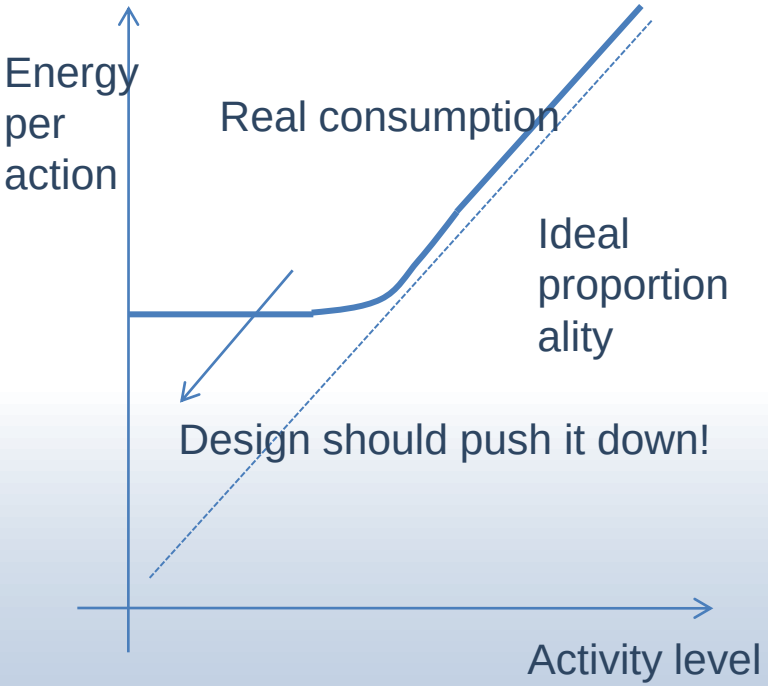
	Constant Throughput/Latency		Variable Throughput/Latency
	Design Time	Non-Active Modules	Run Time
Dynamic & Short Circuit	Logic Re-Structuring, Logic Sizing, Reduced V_{DD}, Multi-V_{DD} **2.5X**	Clock Gating **2X**	Dynamic or Adaptive Frequency & Voltage Scaling **2.5X**
Leakage	Stack Effect, Multi-V_{TH} **2X-10X**	Sleep Transistors, Multi-V_{DD}, Variable V_{TH} **10X-1000X**	Variable V_{TH} **2X-10X**

Source: J. Rabaey, UCB 2005

Newcastle University

holistic energy harvesting

Energy-proportional computing



"Systems tend to be designed and optimized for peak performance. In reality, most computation nodes, networks and storage devices typically operate at a fraction of the maximum load, and do this with surprisingly low energy efficiency. If we could design systems that do nothing well (as phrased by David Culler), major energy savings would be enabled. Accomplishing energy-proportional computing requires a full-fledged top-down and bottom-up approach to the design of IT systems." (from Jan Rabaey's lecture **The Art of Green Design: Doing Nothing Well** – March 2010**)**

Portable Power Supplies

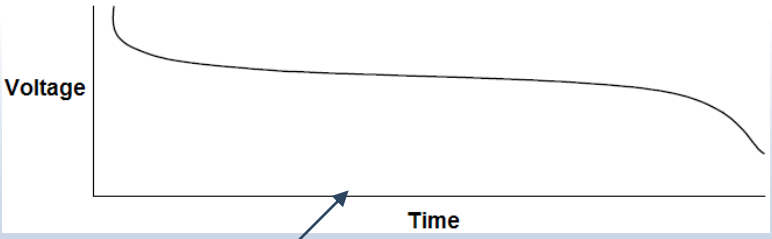
For mobile computing applications the choices of power supply are either batteries or emerging energy-harvester supplies.

Battery

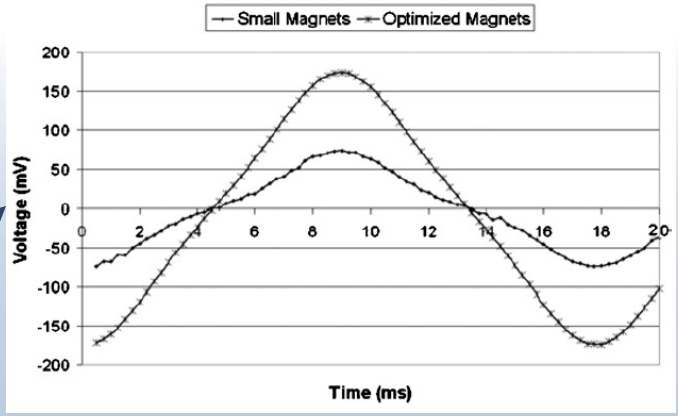
- Can supply finite energy (E) – depends on the battery capacity.
- The available power (P) can be very large.

Energy-Harvester

- Can supply infinite energy (E).
- The rate of energy production (dE/dt = P) is variable and can be small.



Typical Battery Discharge Curve



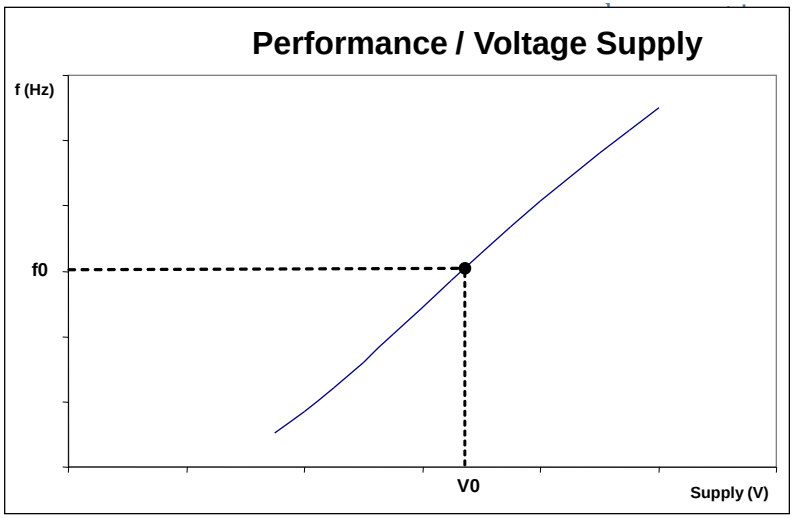
A Micro-Electromagnetic vibration harvester output voltage

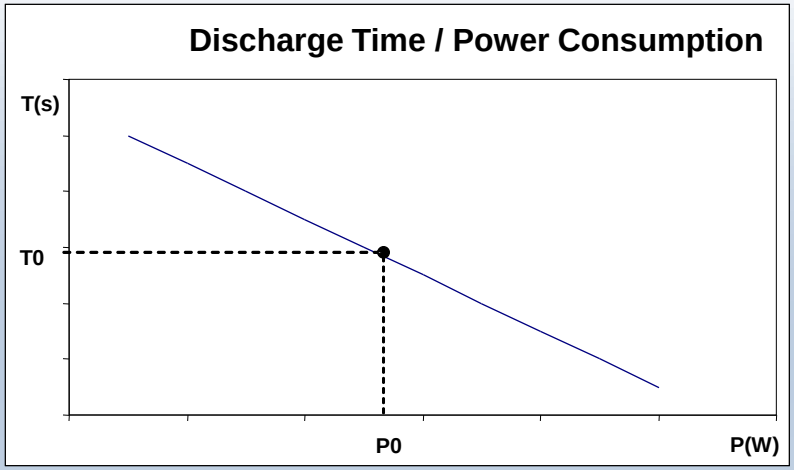
S P Beeby et al., 2007, "A micro electromagnetic generator for vibration energy harvesting", J. Micromech. Microeng. 17 (2007) 1257–1265.

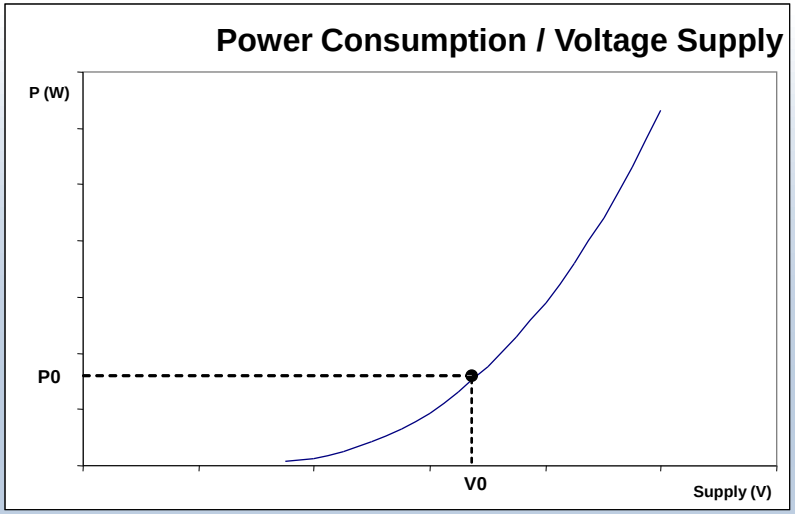
Battery Supplied Circuits

holistic

- Specifications determine the required operating time for the circuit (T0)
- Available energy E is constant so T0 determines power consumption of the circuit
- Supply characteristics stable and known in advance
- Consumption depends on the computational load and may vary

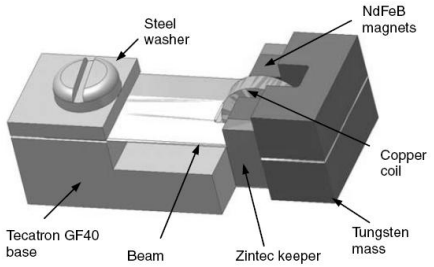




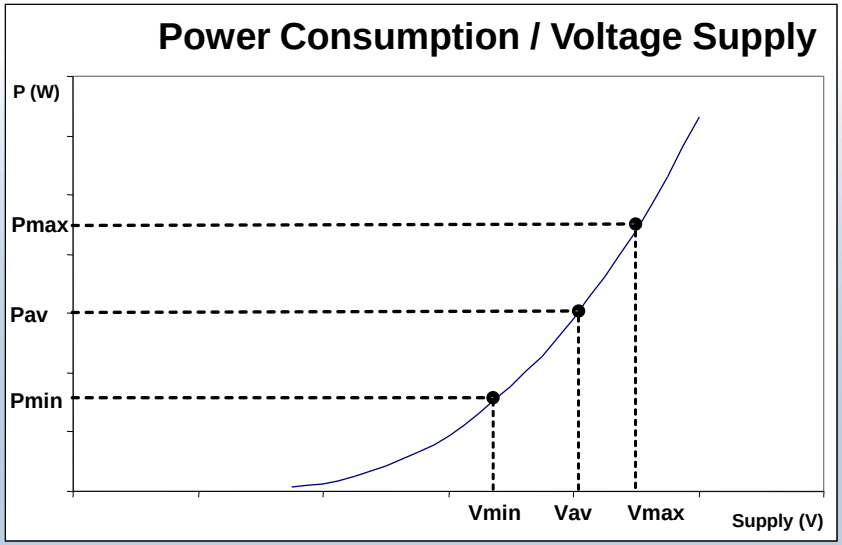


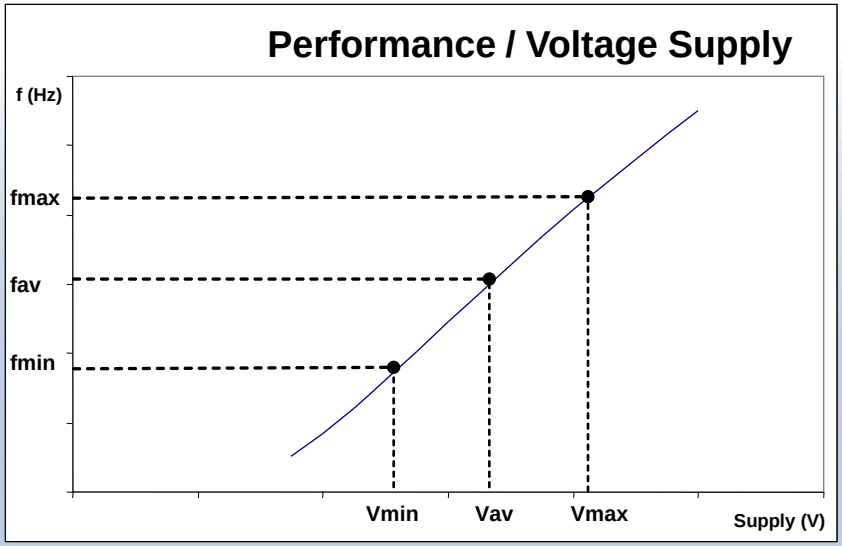
Energy-Harvester Supplied Circuits

- Specifications determine the possible output power range (Pmin, Pmax)
- Power P is variable depending on ambient conditions
- Supply characteristics may be unstable and unpredictable
- Consumption modes may be different, but for simple sensor systems the load is simple and regular, so scheduling computations to modulate supply is required



S P Beeby et al., 2007, "A micro electromagnetic generator for vibration energy harvesting", J. Micromech. Microeng. 17 (2007) 1257–1265.





Circuit Designer Choices (1)

Battery Supply ⇨

- Determine from T0 the required power consumption P0.
- Design the circuit for constant P0 consumption → constant V0 supply → constant f0 performance (or apply DVS and DVFS to maximise battery life)

Energy-Harvester Supply ⇨

- Design the circuit for constant Pmin consumption → constant Vmin supply → constant fmin performance.

OR

- Track available power Paverage → change circuit consumption/performance in real-time → faverage > fmin.

Circuit Designer Choices (2)

To maximise a circuit's power utilization of a variable power output source:
- increase voltage supply of the circuit to the energy-optimum value (variable voltage).
- switch on/off parts of the circuit (constant voltage).

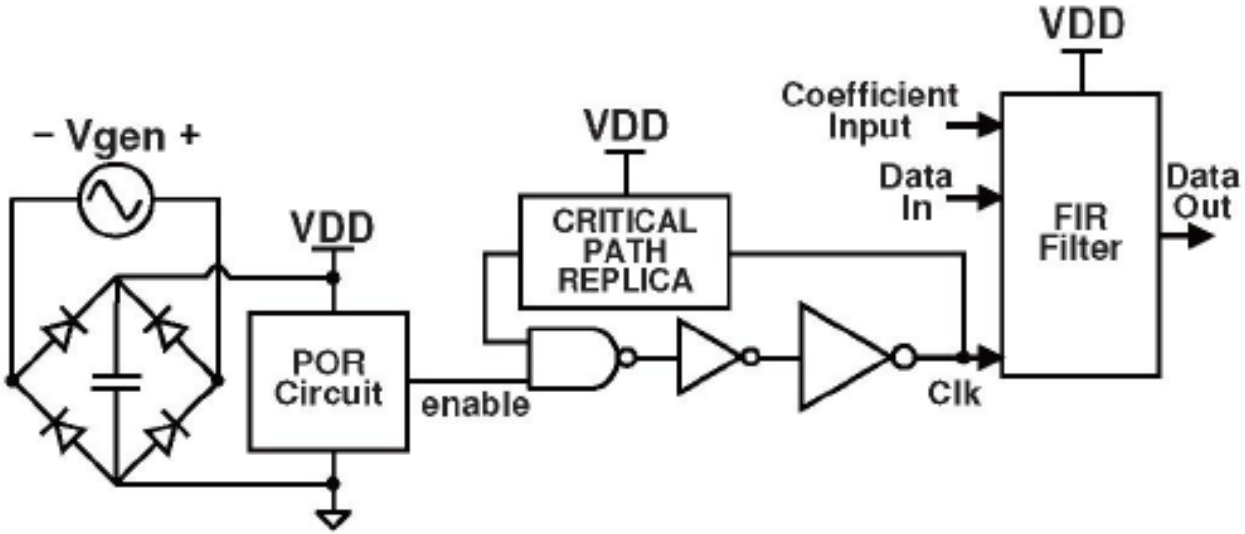
Real-time ⇦

- For both cases special controller circuits have to be developed.
- For the first case (variable voltage) self-timed circuits have an advantage → no additional circuit required to change the operating frequency.

AC supplied self-timed circuits have been demonstrated in practice.

For every power supply cycle: wake up the circuit, perform computation and shut down the circuit – hence, power-on reset needed.

AC-powered self-timed circuit

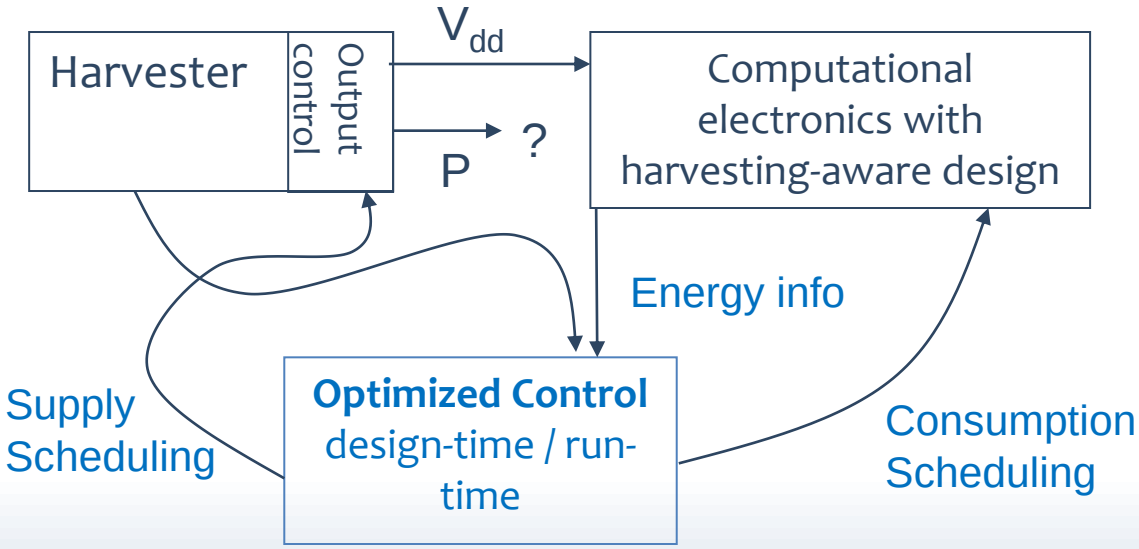


Fast Power-on Reset (4.1nW),
3T DRAM to keep state across supply cycles,
135K transistors in 180nm CMOS
Can supply 250KHz on all process corners for <=50C

J Wenck, R Amirtharajah, J Collier and J Siebert, 2007, "AC Power Supply
Circuits for Energy Harvesting", 2007 IEEE Symposium on VLSI Circuits, 92-93.

Problems: critical path replica may not scale well with the computational load (cf. SRAM delay matching problems – following slides)

Power-adaptive Computing (Holistic view)



Optimization

$$\frac{E_C}{E_S} = Max$$

Useful energy consumption is maximized for a given amount of energy produced , or

Energy supplied is minimized for given amount of energy consumed usefully (to carry out specified/required computation)

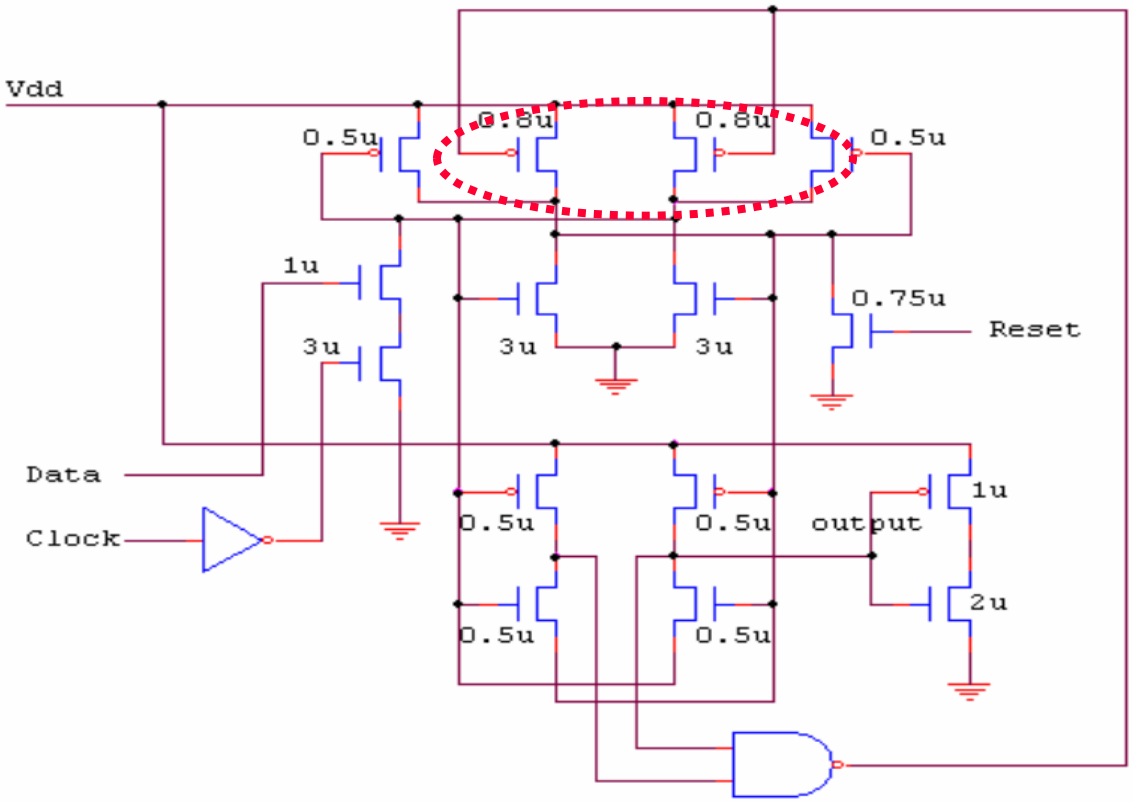
Power-Adaptive System Design

- Adaptation levels:
  - Cell and component level
    - Resilience to Vdd variations (e.g. robust synchronisation, self-timed logic and completion detection)
    - Leakage control mechanisms (e.g. body biasing)
  - Circuit level (clock/power gating, DVF scaling)
  - System level (power sensing and control of power supply and consumption chains)
    - Optimal control of Vdd for minimum energy per operation
    - Control of computation load to fit the power profile or optimise for average power

Asynchronous (self-timed) design principles improve effectiveness and efficiency of both sensing and control in adaptation process

Robust Synchronizer (adapting performance to Vdd changes)

This circuit turns on extra power when in meta-stable state and turns off after that

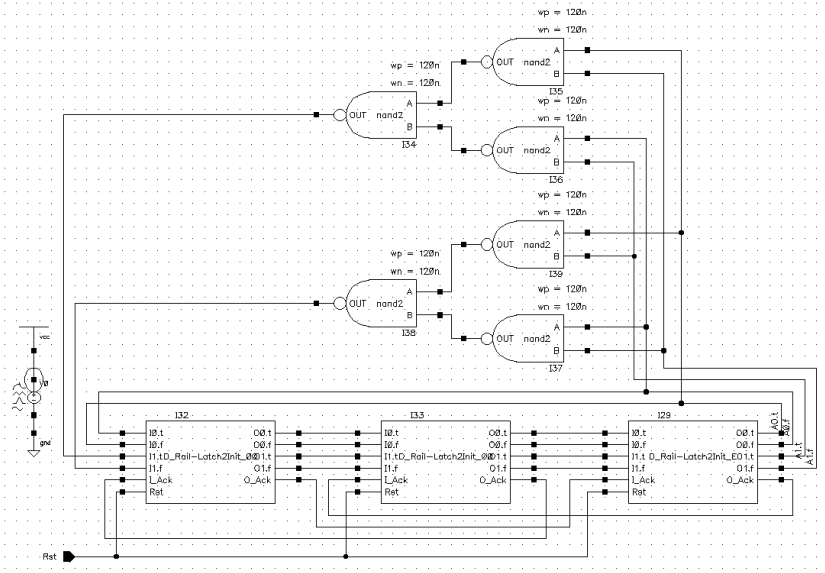


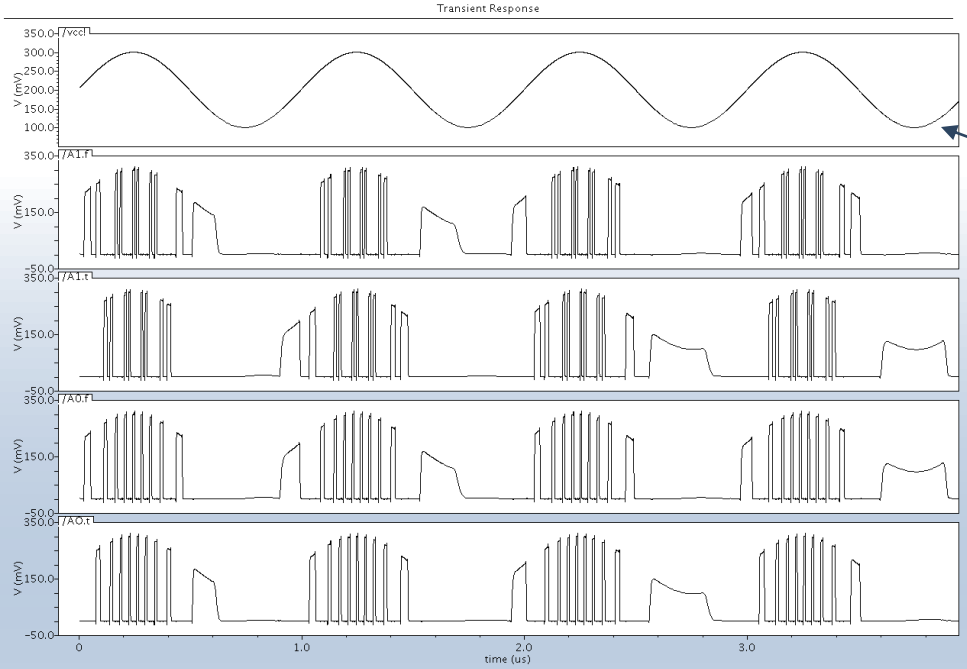
Can be further improved, to enable work at subthreshold Vdd, by means of body biasing of all main transistors

Source: J. Zhou et al, Newcastle, 2007

Closer look at AC-powered self-timed logic

2-bit Sequential Dual-rail Asynchronous Counter





A1.f

A1.t

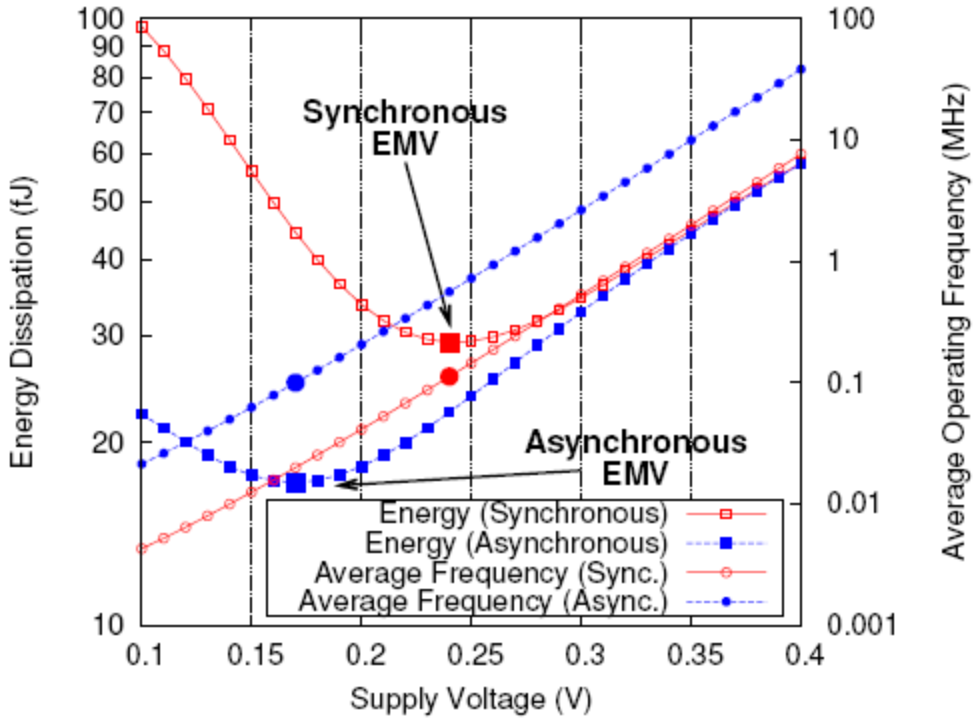
A0.f

A0.t

Supply: AC 200mV±100mV
Frequency: 1Mhz

Self-timed logic with completion detection is robust to power supply variations

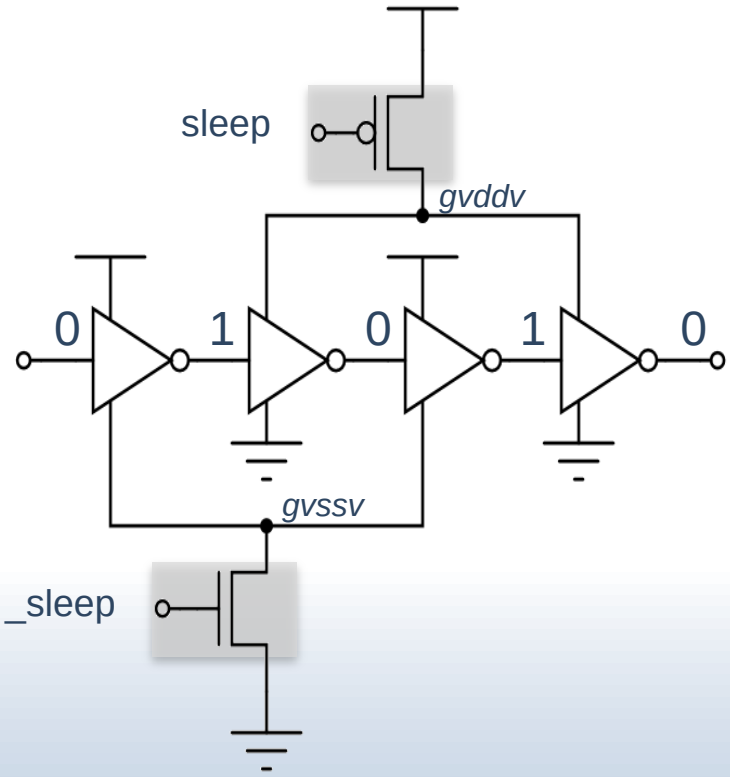
Synchronous vs Asynchronous Design (in terms of energy efficiency)



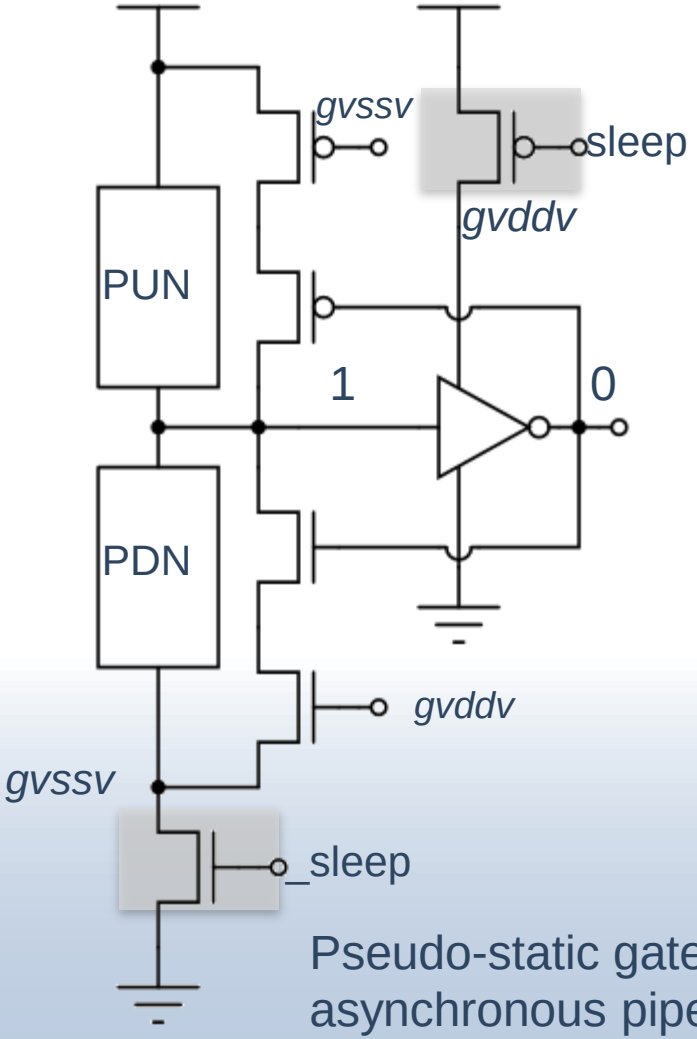
Asynchronous (self-timed) logic can provide completion detection and thus reduce the interval of leakage to minimum, thereby doing nothing well!

Source: Akgun et al, ASYNC'10

Asynchronous (State Preserving) Power Gating



Standard gates for combinational logic

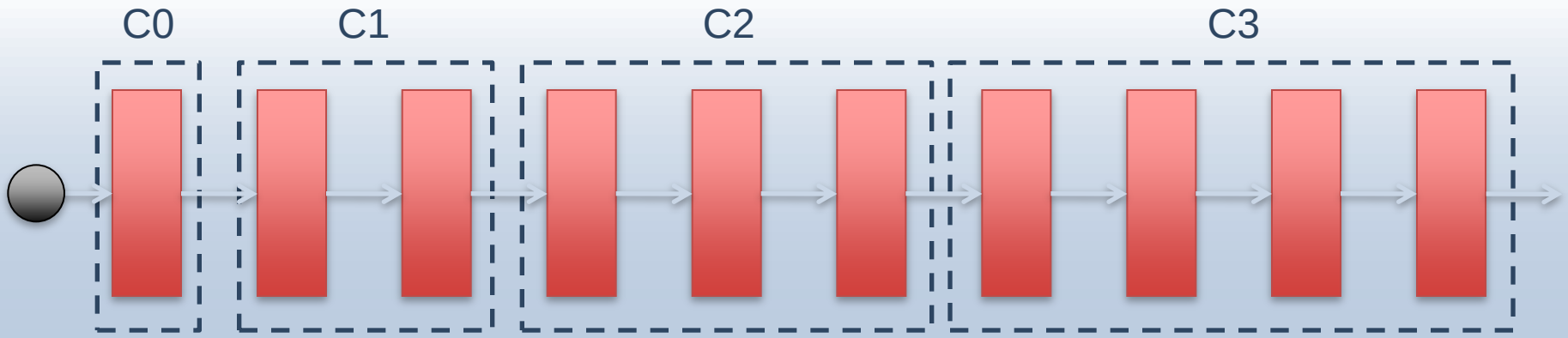


Pseudo-static gates for asynchronous pipelines

Source: Ortega et al, ASYNC'10

Zero-Delay Ripple Turn On

- Leverage Pipeline Stage Computation Latency
  - Hide Latency of Powering Up Downstream Stages
- Leverage Asynchronous Circuit Robustness
  - Do Computation During Power Up
- Pipeline Cluster = Power Gating Domain
- Choose
  - Pipeline Cluster sizing for your application
  - Power Gating techniques for each cluster



Pipeline Clusters

Source: Ortega et al, ASYNC'10

Newcastle University

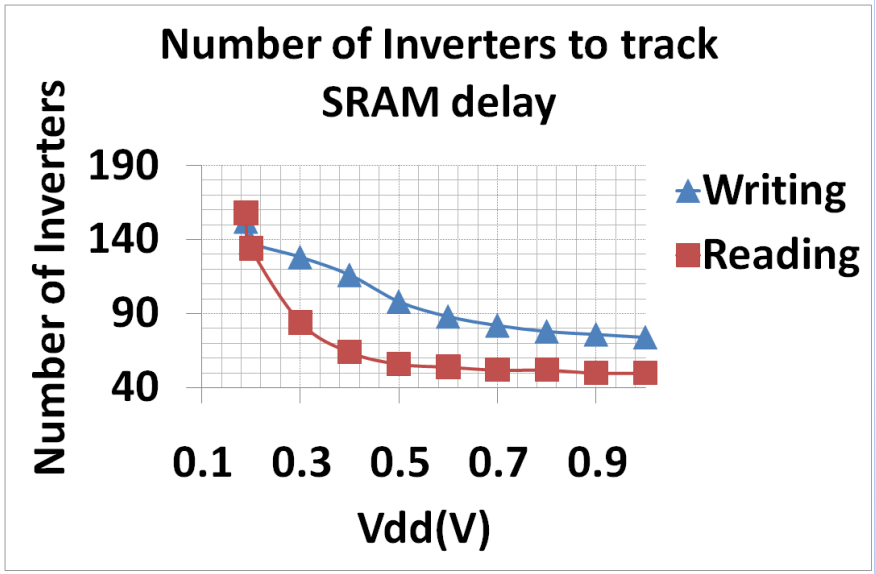
holistic energy harvesting

Focus of our research in Holistic Project

- Component level characterisation and design:
  - Inverter chain, ring oscillators, counters, arithmetic, SRAM, DRAM cells
  - Design of self-timed (sub-threshold) logic
- Power control methods:
  - New power gating techniques to reduce leakage in computational load for lower frequency range
- Power-adaptive system design:
  - Supply and consumption modelling and control
  - Power Sensing
- System-level power management:
  - Statistical modelling and analysis

Delay Mismatch in existing asynchronous (bundled delay) SRAMs

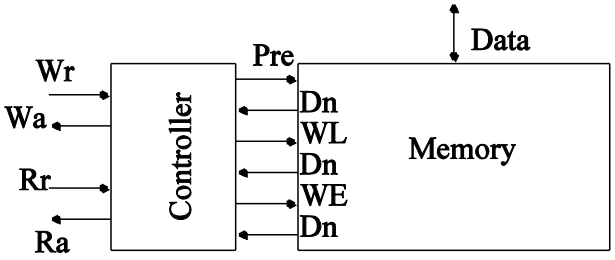
- Mismatch between delay lines and SRAM memories when reducing Vdd



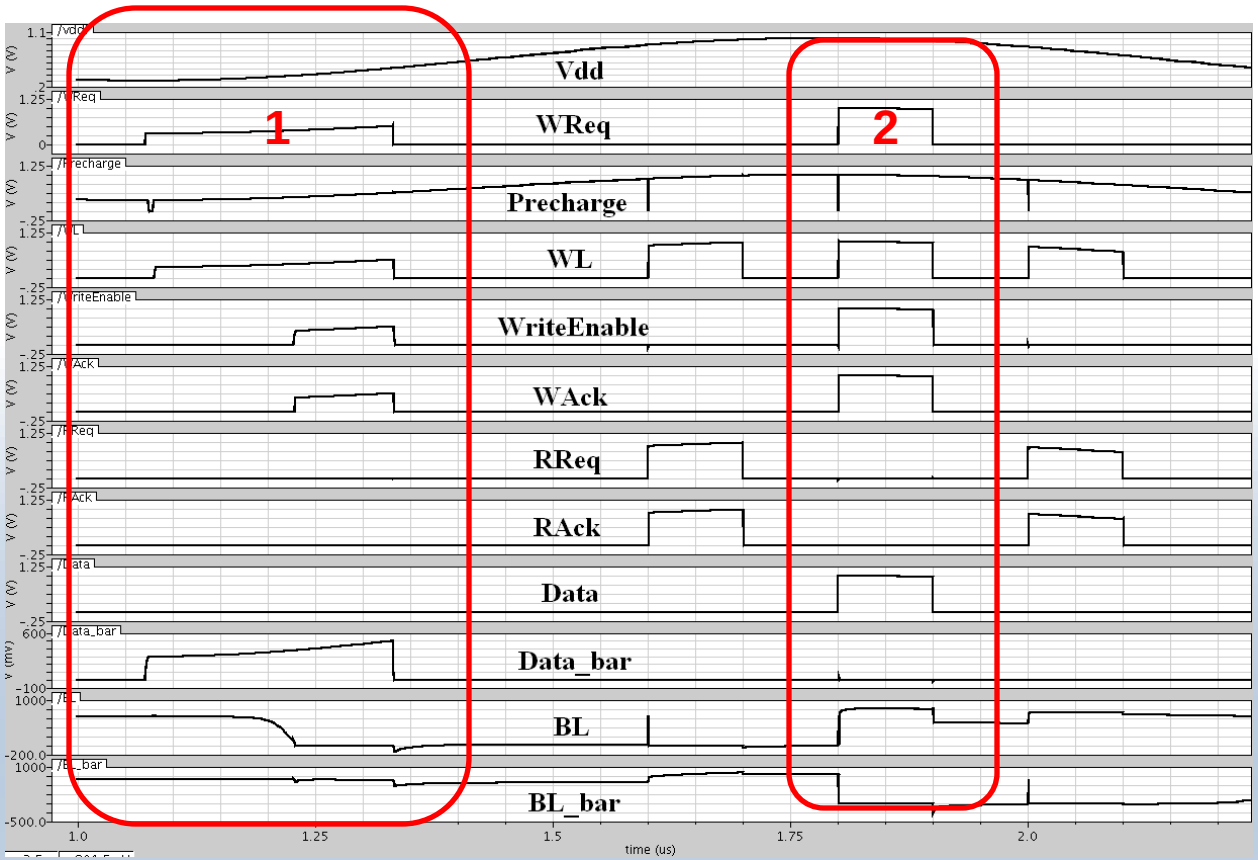
For example, under 1V Vdd, the delay of SRAM reading is equal to 50 inverters and under 190mV, the delay is equal to 158 inverters

- The problem has been well known so far
- Existing solutions:
  - Different delay lines in different range of Vdd
  - Duplicating a column of SRAM to be a delay line to bundle the whole SRAM
- The solutions require:
  - voltage references
  - DC-DC adaptor
- Completion detection needed?!

SRAM: Speed Independent Solution



The SI controller uses completion detection in SRAM and handshake protocols to manage pre-charge, WL and WE in the SRAM banks

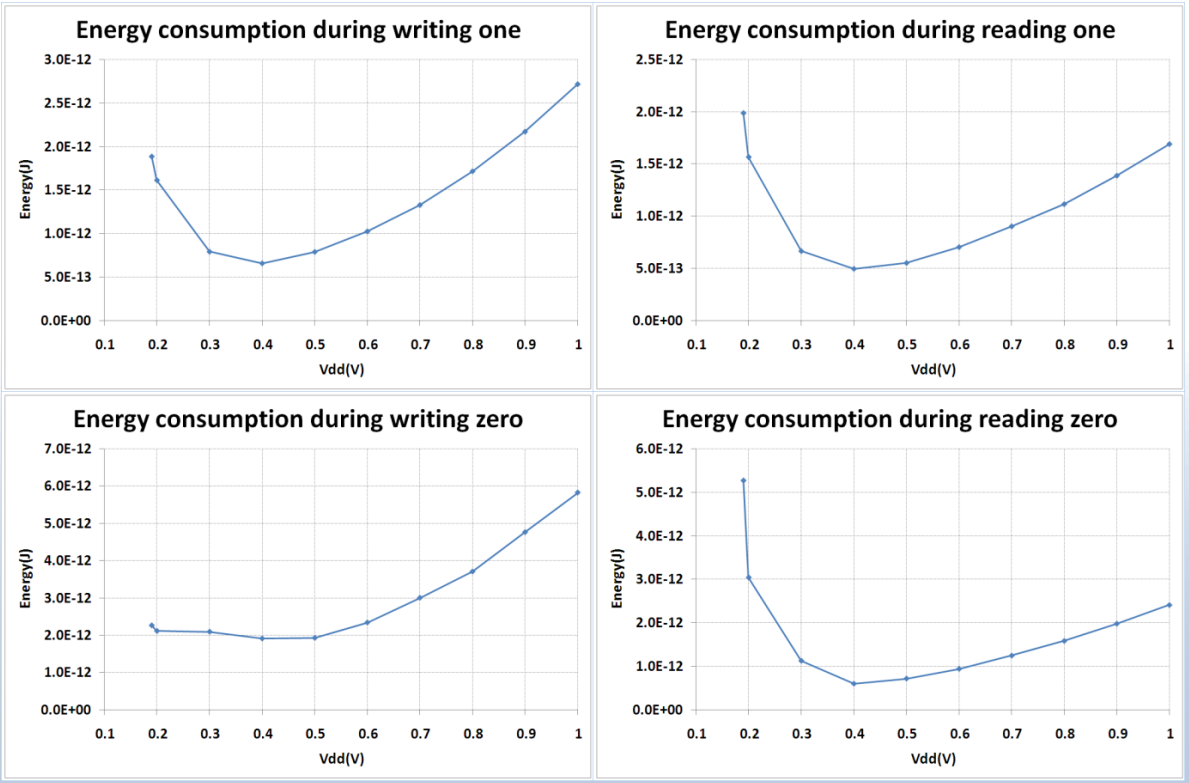


Can work smoothly under variable Vdd.

For example, the first writing works in low Vdd, it takes long time, and the second writing works in high Vdd, works faster.

New Speed-independent SRAM

1k-bit (64x16) SI SRAM is implemented using the Cadence toolkit with the UMC 90nm CMOS technology

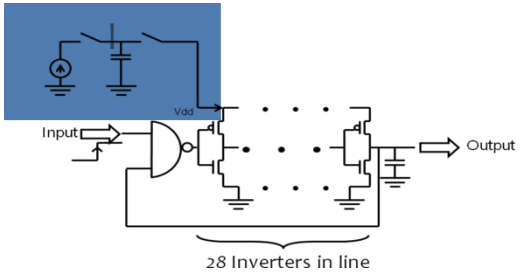


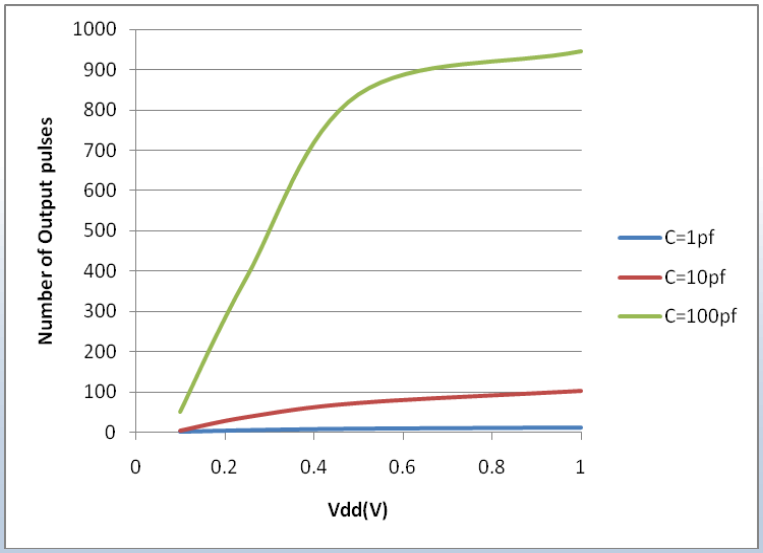
The curves show that the minimum energy point of the chip is at 400mV-500mV.

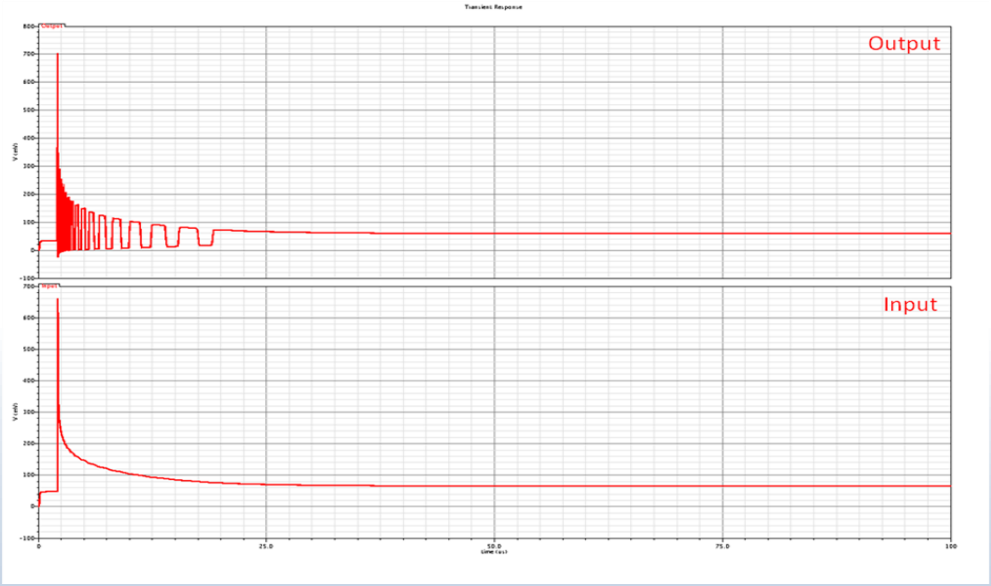
The SRAM consumes 5.8pJ in 1V when writing a 16-bit word to the SRAM memory and 1.9pJ in 400mV.

Power Sensing via self-timing(1)

Varying Vdd supply: "computation model" with limited energy and power source

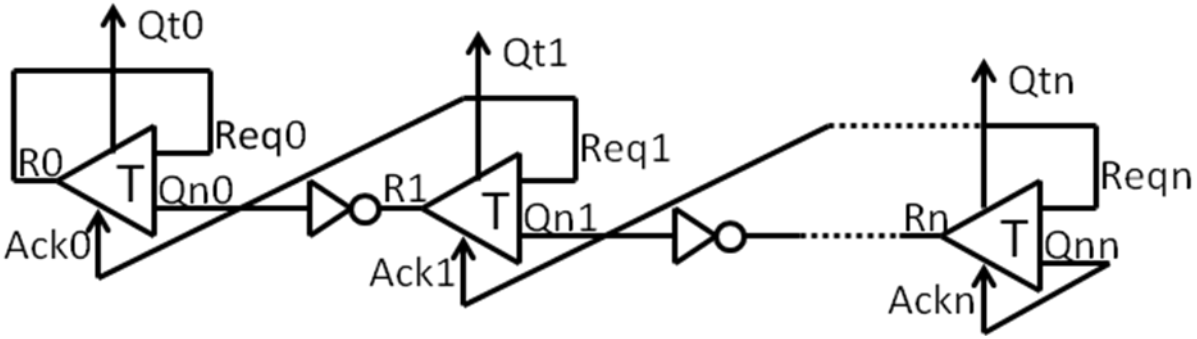






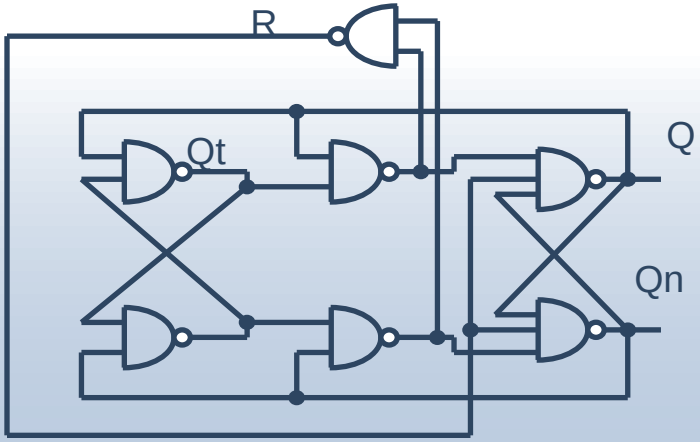
Power Sensing via self-timing(2)

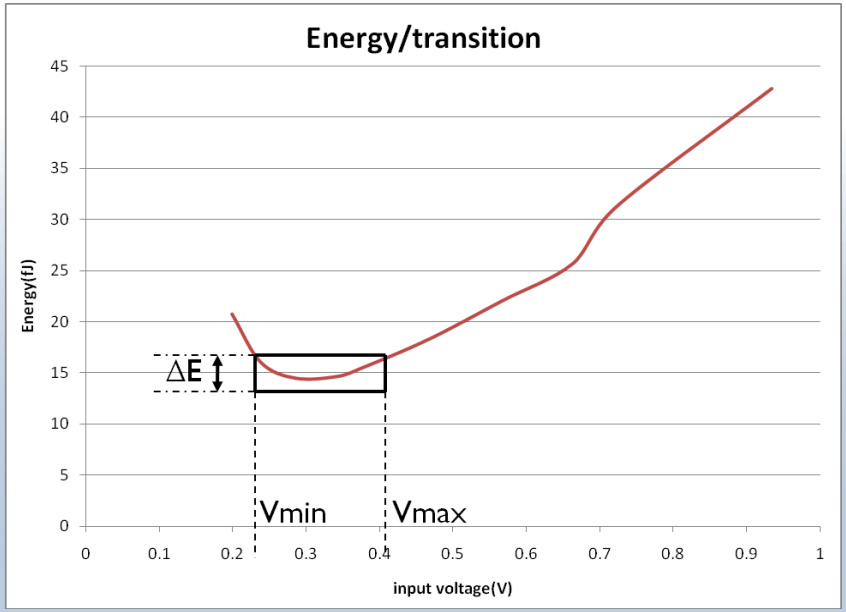
6-bit Self-timed Counter



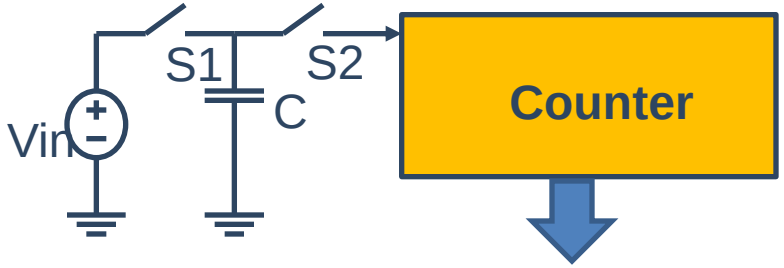
Energy optimality region

Toggle

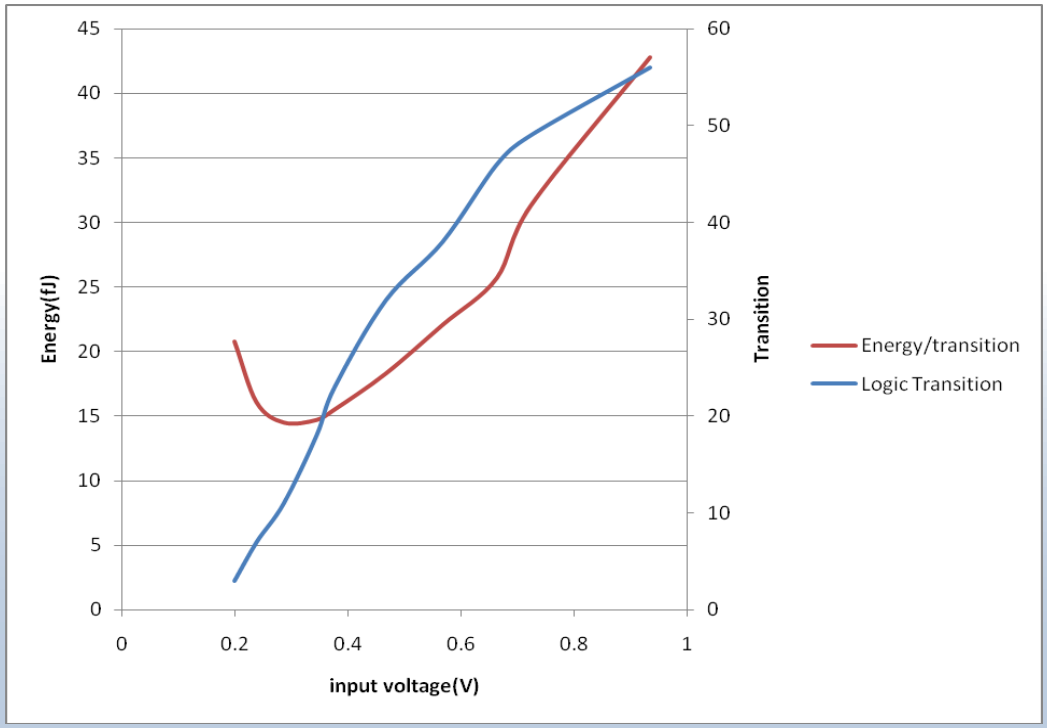


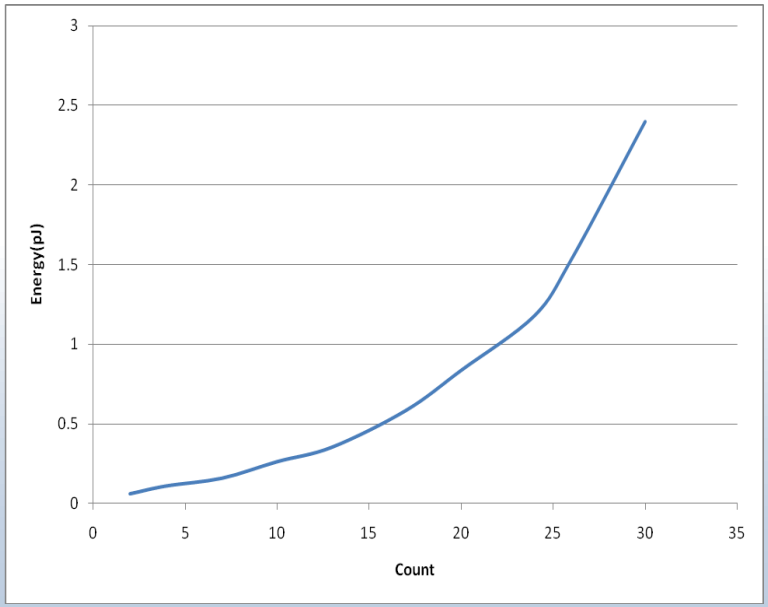


Power Sensing via self-timing (3): Charge to code conversion



Energy and transition count vs different Vdd samples into the Capacitor





UK Patent application 1005372.6 (30.03.10), Newcastle University

Newcastle University

Conclusions

holistic energy harvesting

- Energy-harvesting changes the dynamic balance between supply and consumption – supply add operational constraints in real-time
- Adaptation to power changes should be at all levels of abstraction, from logic cells to systems
- Asynchronous (self-timed) techniques support more effective adaptation to Vdd changes via natural temporal robustness; they also offer better energy proportionality
- Good energy characterisation of loads (logic, memory, i/o, RF) is essential for high-quality adaptation
- More theory, models and algorithms are needed for handling the problem of power-adaptation in run-time (work also started at Newcastle on computation models with energy tokens)

Acknowledgements

Members of “Microelectronics Systems Design” research group at Newcastle: Panagiotis Asimakopoulos, Alex Bystrov, Terrence Mak, David Kinniment, Andrey Mokhov, Delong Shang, Danil Sokolov, Fei Xia, Reza Ramezani, Zhou Yu, Abdullah Baz, Xuefu Zhang, involved in the Power-Adaptive Computing research

EPSRC funding (EP/G066728) – Project “Next Generation Energy-Harvesting Electronics: A Holistic Approach”, involving Universities of Southampton, Bristol, Newcastle and Imperial College
(see Paul Mitcheson talk later today)

Newcastle University

Run-time power modulation by dynamic scheduling methods

holistic energy harvesting

- Objectives

Modulate the power consumption of a system which is constrained by real-time power supply, e.g. in an energy-harvesting-system (EHS), by tuning the concurrency degree of the system by dynamic scheduling methods, such that the power consumption of the system will satisfy the power supply bounds, and at the same time, achieve certain optimality in performance (e.g., its execution latency).

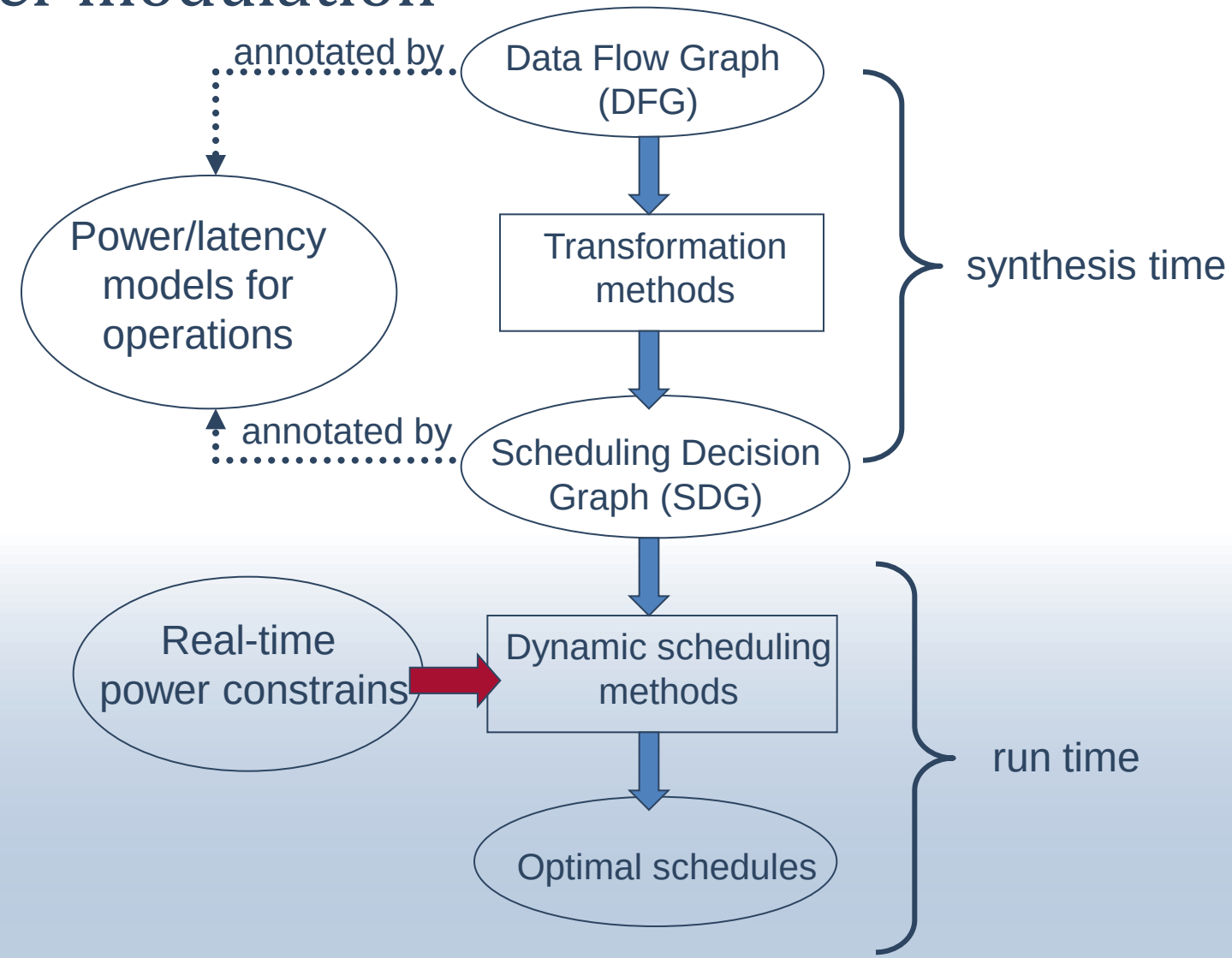
- Rationale for power modulation

Adjusting the concurrency degree of a system by tuning of the active capacitance for charge/discharge, according to the dynamic power consumption formula $P = \alpha C f V^2$

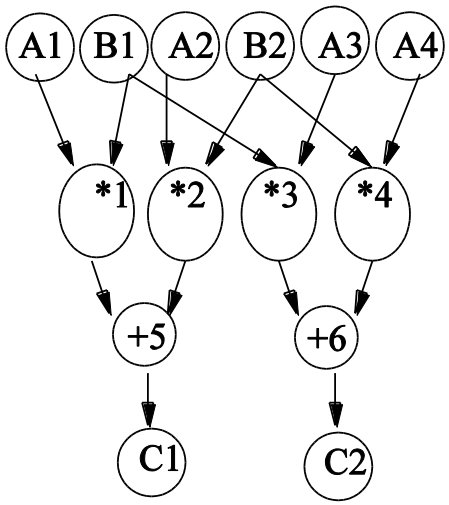
- Effects on power consumption compared with other methods

cf: voltage scaling (quadratic on adjusting power), frequency scaling (clock cycling) etc.

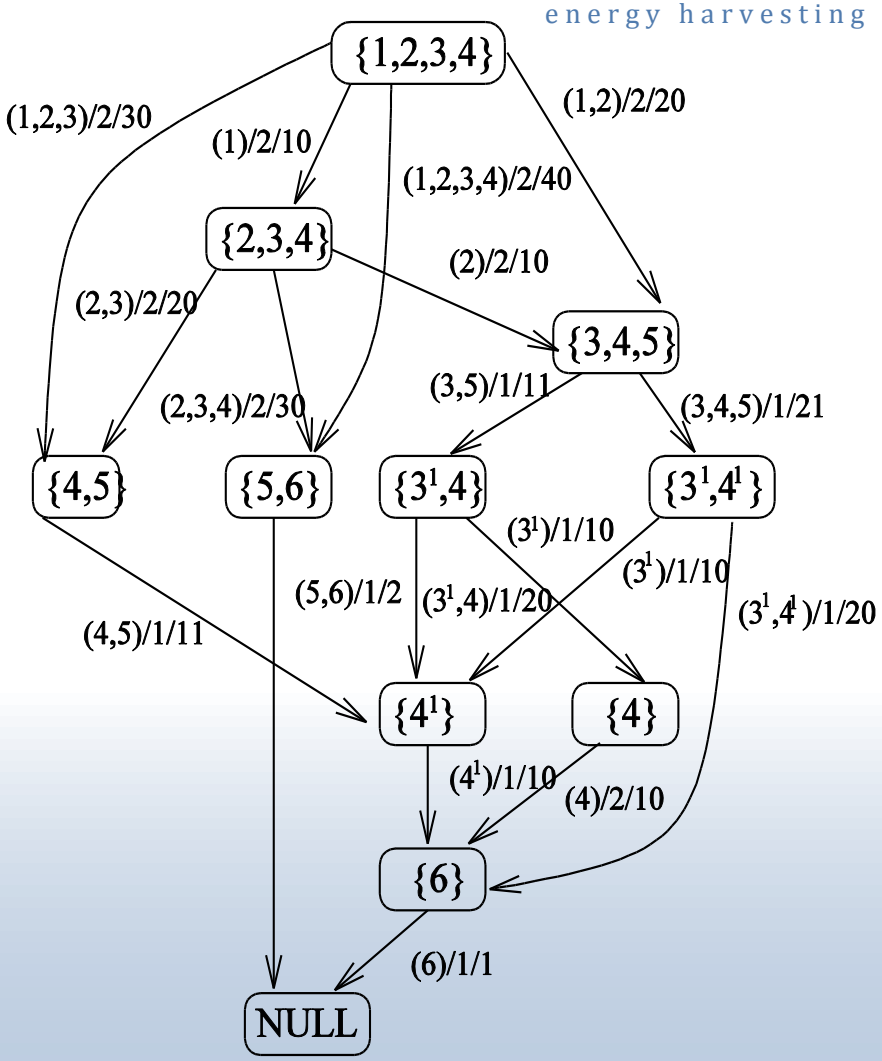
A design flow for run-time power modulation



A truncated transformation from a DFG to its SDG



DFG-2-SDG



A toy matrix multiplication example including

additions (op 5~6):

1 unit delay and 1 unit power

multiplication (op 1~4):

2 units delay and 10 units power

Scheduling decision graph (SDG), transformation methods, and scheduling policies

- A SDG is a triplet (V, E, F) where

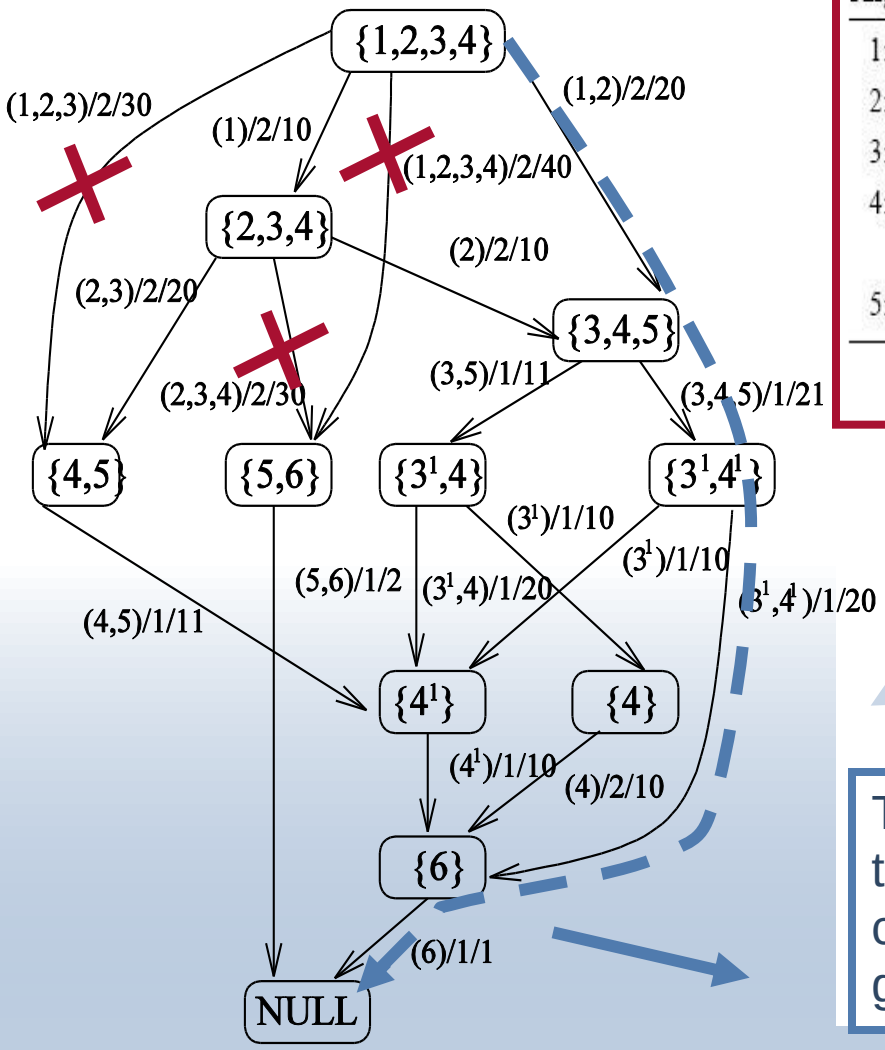
  V is the vertex set, and each vertex is a state when scheduling the DFG and is labeled by the operation set ready for scheduling at that state.

  E is the edge set, and each edge represents a *schedule step* at a state. A step is labeled with triple elements: the operations scheduled in the step, its length (in terms of clock cycles devoted to executing the step), and the associated power.

  F is the flow relation specifying how a state enables a scheduling step.
- A schedule corresponds to a path from the initial state to the Null state.
- Algorithms exist for both complete and truncated transformation from DFG to SDG.
- Scheduling policies for a truncated transformation for now consider the following two constraints:

  1) Concurrency degree for an operation type - how many operations belonging to that type can be scheduled during a step.
  2) Combination of the operations belonging to a certain type.

A dynamical scheduling algorithm for run-time power modulation



Algorithm run-time heuristic scheduling with SDG

1: **Inputs**: A $SDG = (V, E, F)$ and $P(t_s)$
2: **Outputs**: An optimal schedule for t_s
3: **prune** the edges e in SDG with $p(e) > P(t_s)$
4: **find** the path s in the pruned SDG with the shortest latency, i.e., $min(\sum_{i=0}^{n} \lambda(e_i) : e_i \in s)$
5: **form** the schedule according to s

The optimal path has a minimal latency of 5 time units and a maximal average power consumption of 16.4 units, in the remaining graph.