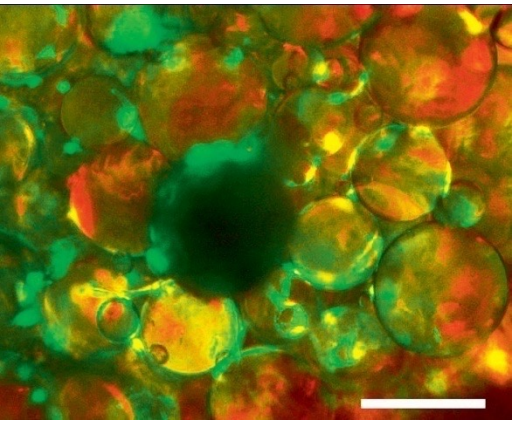


Partially-Reconfigurable SoC Tiles for Dynamic Power Savings on an FPGA

Michael Grieco
Biruk Seyoum
May 16th, 2025

COLUMBIA | ENGINEERING
The Fu Foundation School of Engineering and Applied Science

Motivation

- ASIC-based SoCs with power management
  - Demand-based allocation with DVFS

Blitzcoin

- More limited on an FPGA
  - Dynamic Frequency Scaling
  - Dynamic Partial Reconfiguration
  - Clock gating

invokes

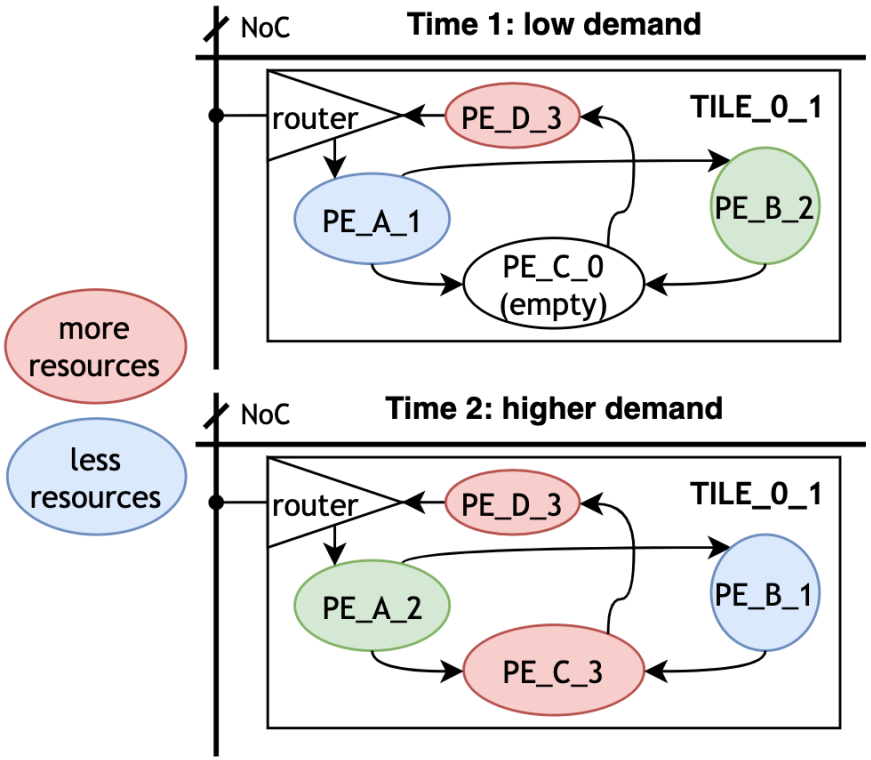
- Power gating
- Clock gating

Controllable
(design flow or runtime)

Technology-based
(static or runtime)

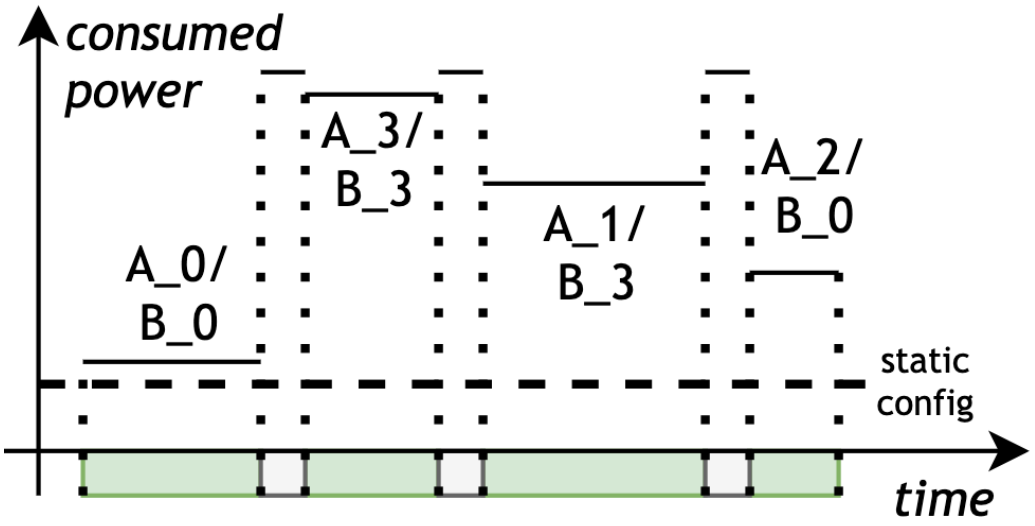
Larger Goals

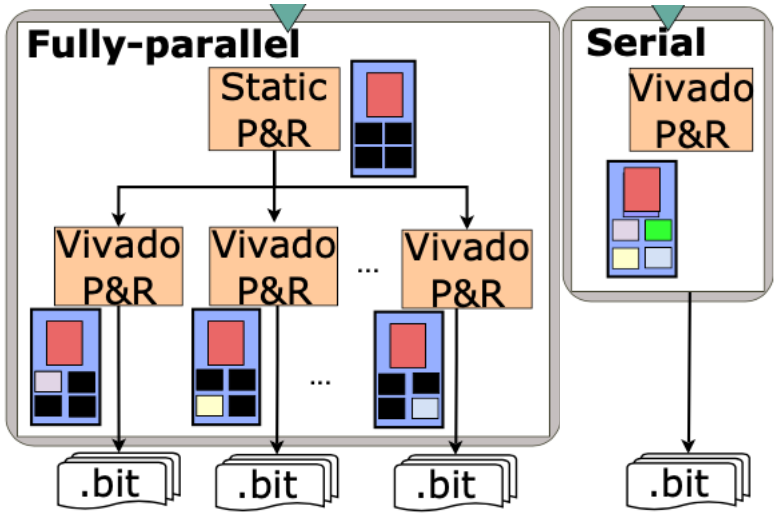
- Demand-based resource allocation
- Fine-grained reconfiguration
  - More design flexibility (compilation)
- Fine-grained power management when combined with clock management



Larger Goals

- Smaller regions => smaller bitstreams => less reconfiguration time/faster P&R

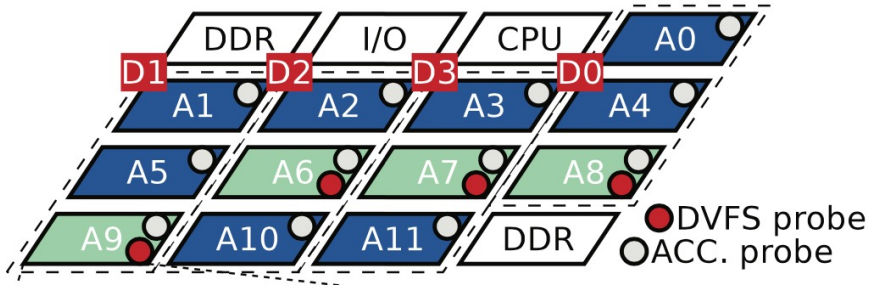




(Seyoum et al., 2023)

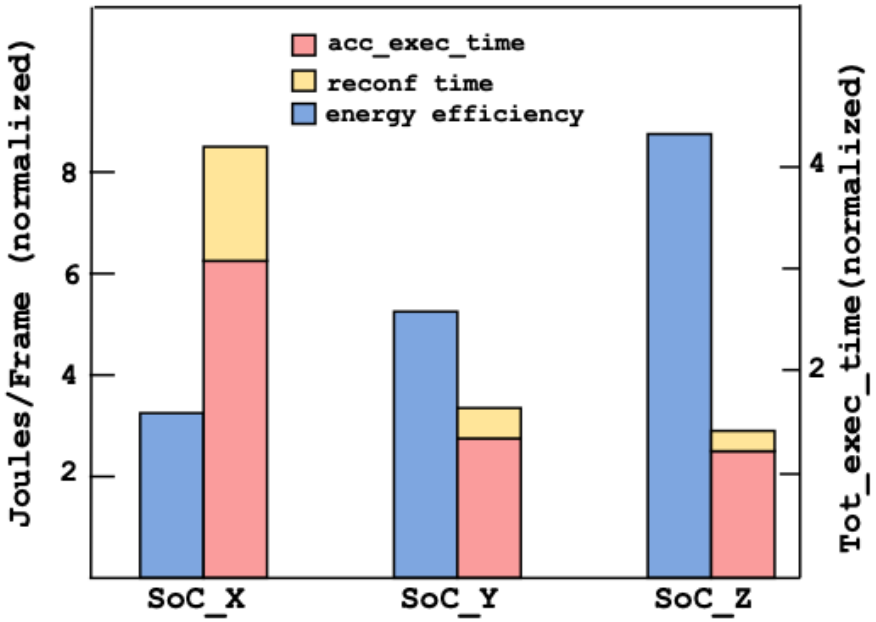
Initial Contributions

Tile-Based DFS (Mantovani et al., 2016)



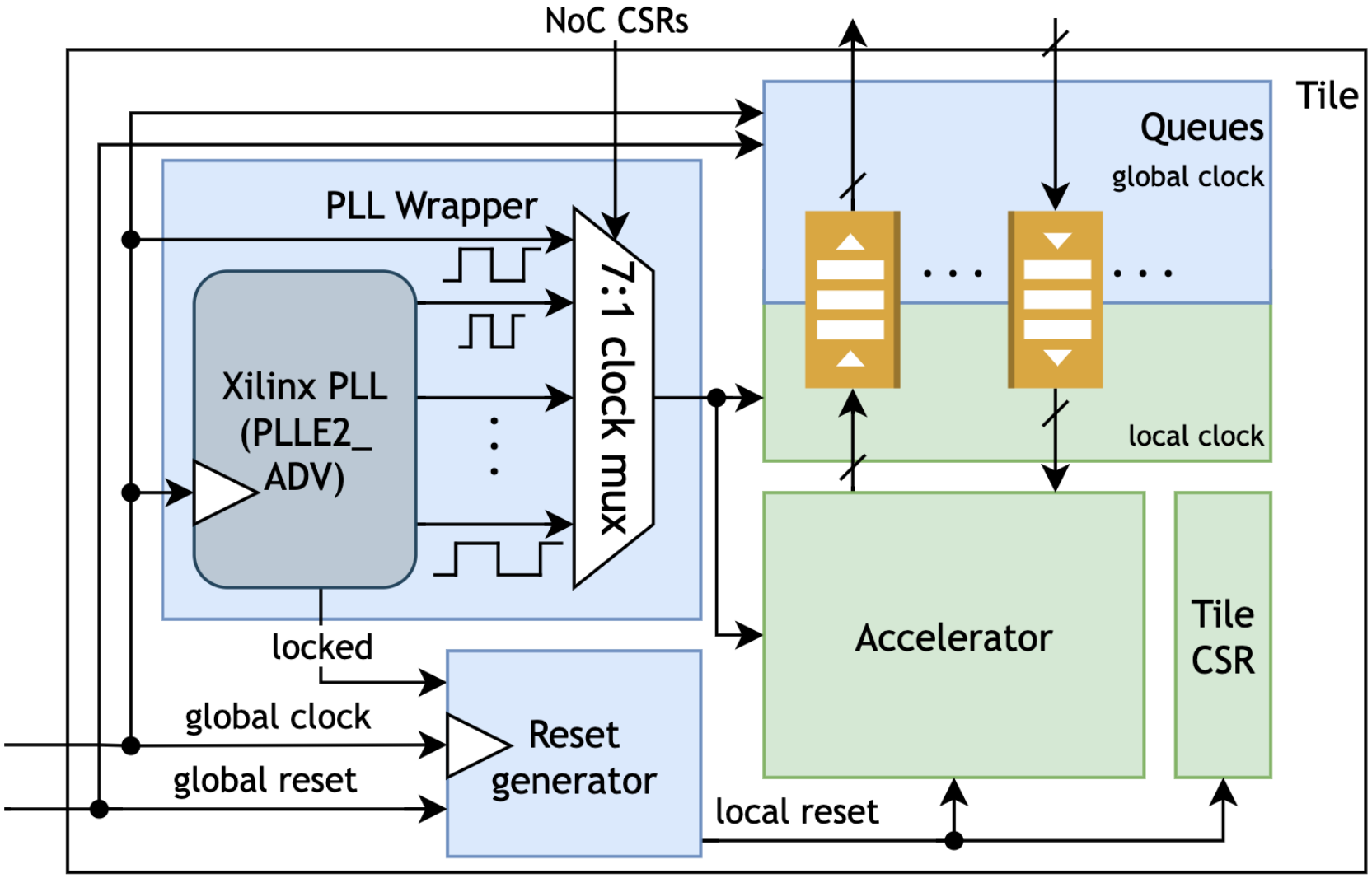
- Existing work for ASIC
- Implemented an FPGA-based PLL

Tile-Based DPR (Seyoum et al., 2023)

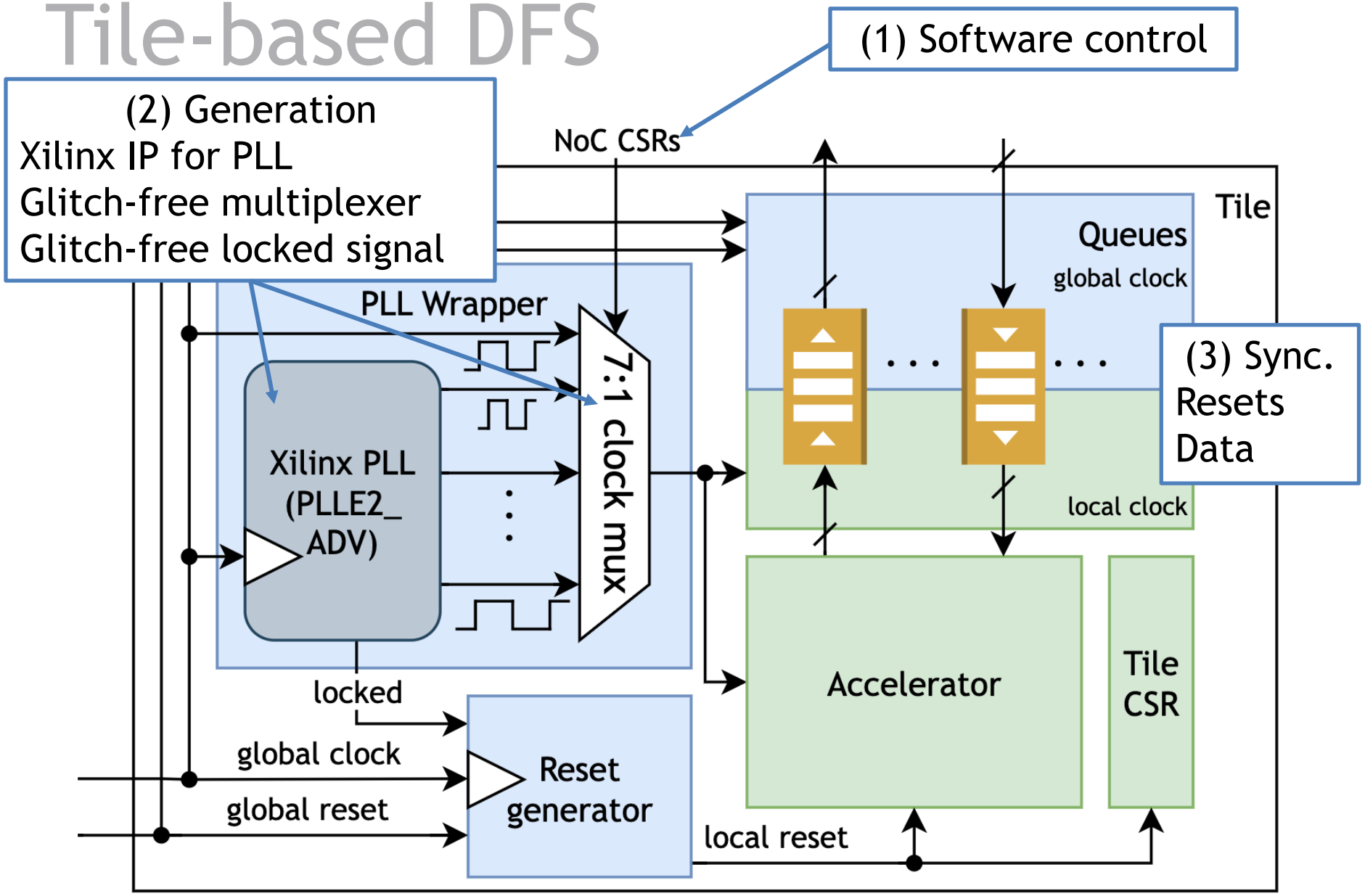


- Existing work is “old”
- Merged with new ESP

Tile-based DFS



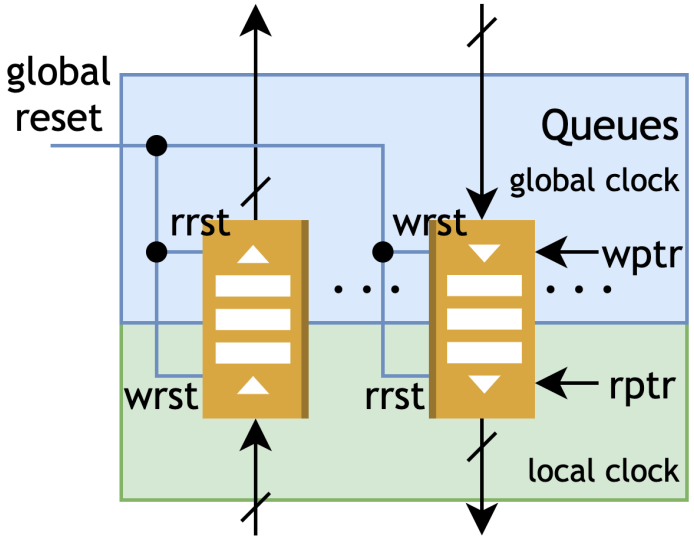
Tile-based DFS



Synchronization challenges

Tile queues

- Read and write asynchronously
- Use global reset for both to sync pointers



CSR reset

- Previous config. gets lost when PLL re-locks
- SW must change PLL then reconfigure

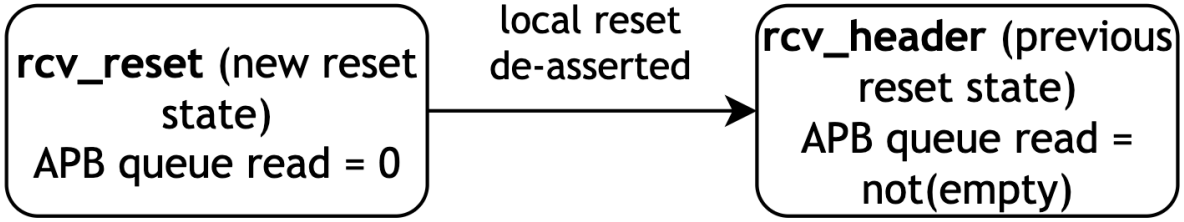
Synchronization challenges

Boot sequence

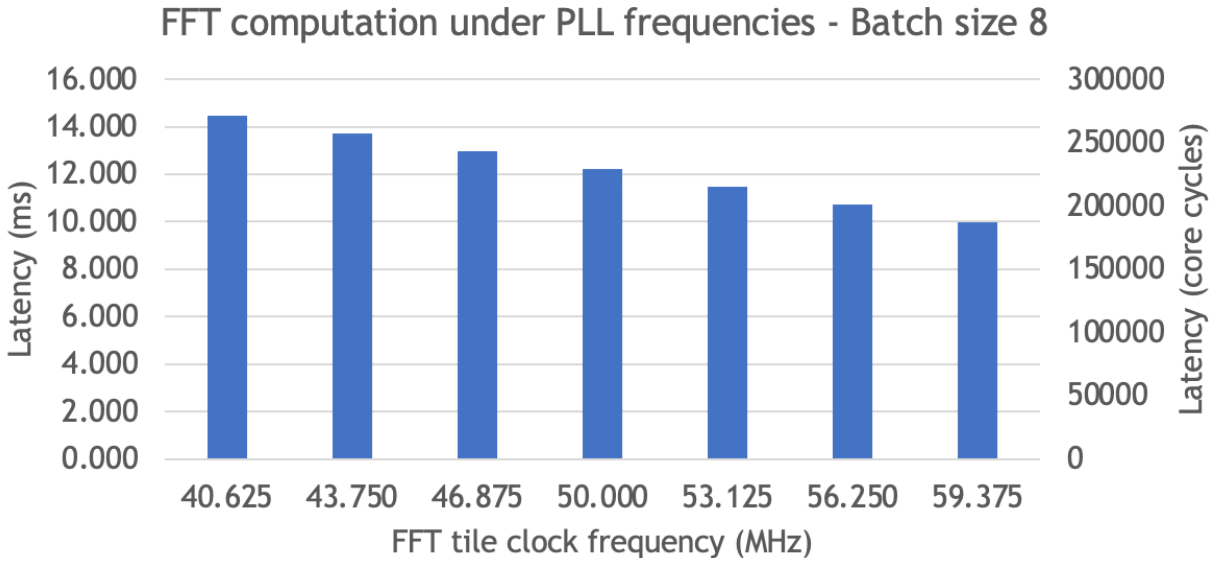
- Tile PLL starts trying to lock once the global clock locks
- Tile's APB proxy read signal: combinational function of "empty"

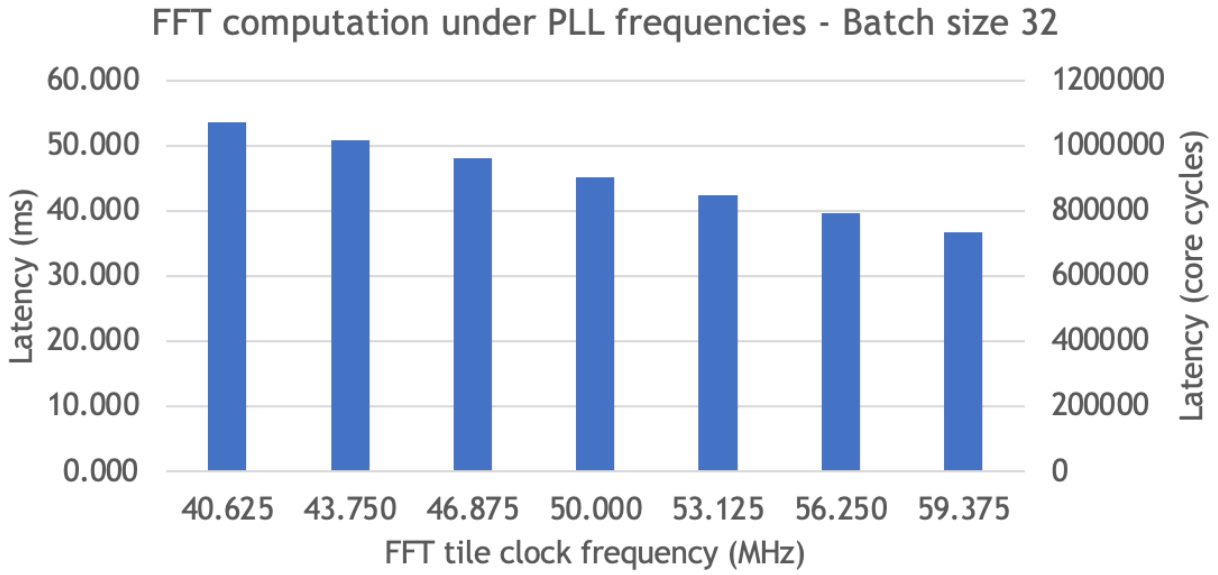
CSR reset (tile_id)

- Boot-written tile ID clears on reset (not the fault of SW)
- ID CSR survives a soft-reset, cleared on hard-reset (global)



DFS Results


FFT computation under PLL frequencies - Batch size 8
Latency (ms)
16.000
14.000
12.000
10.000
8.000
6.000
4.000
2.000
0.000
300000
250000
200000
150000
100000
50000
0
Latency (core cycles)
40.625
43.750
46.875
50.000
53.125
56.250
59.375
FFT tile clock frequency (MHz)

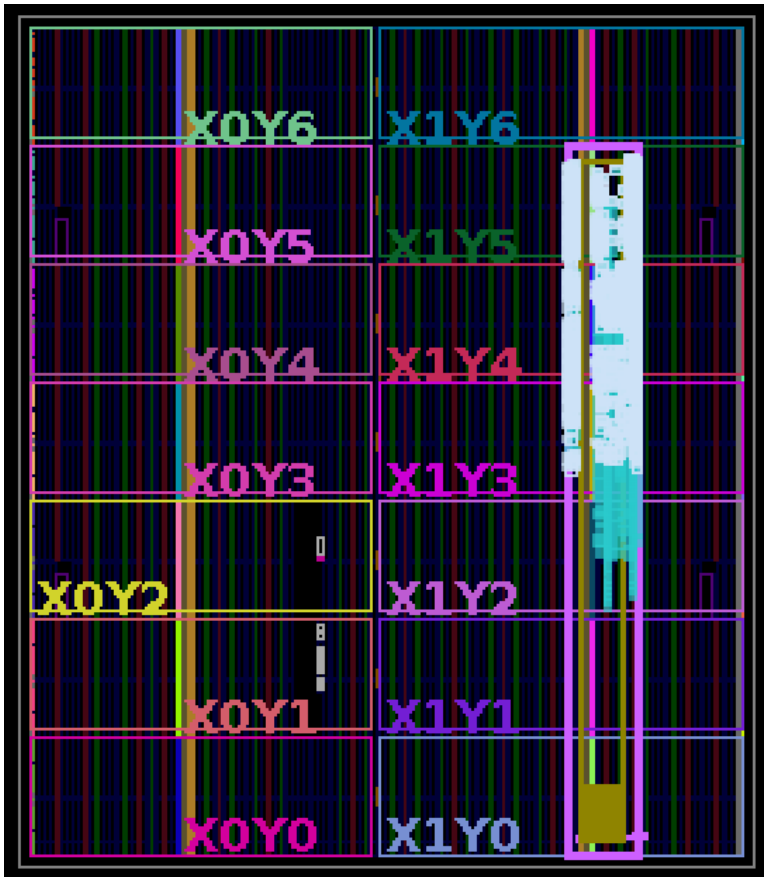

FFT computation under PLL frequencies - Batch size 32
Latency (ms)
60.000
50.000
40.000
30.000
20.000
10.000
0.000
1200000
1000000
800000
600000
400000
200000
0
Latency (core cycles)
40.625
43.750
46.875
50.000
53.125
56.250
59.375
FFT tile clock frequency (MHz)

Partial Reconfiguration

Rebased an "old" branch: DPR examples work in new ESP with Vivado 2023

Added greybox impl.: facilitates power gating (helps static/dynamic)

Out-of-context impl. for power reporting



The OOC power reporting is only robust for dynamic power

Partial Reconfiguration – Power

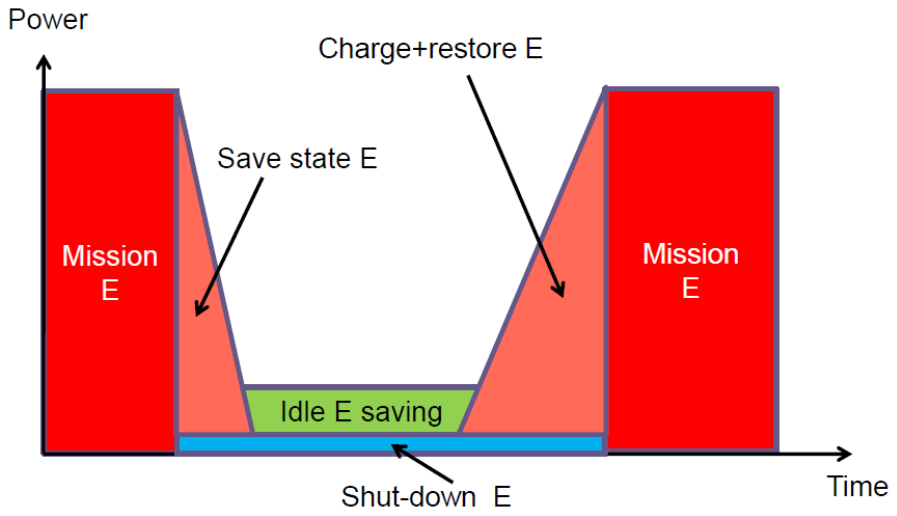
- Break-even time
  - Time to reconfigure: 2 bitstreams
  - Save most dynamic and some static

Unused blocks no longer consume power

Partition pins tied off; no internal gates switching

$$2 * P_{mission} * T_{PRC} \leq (P_{idle} - P_{blanked}) * t_{blanked}$$

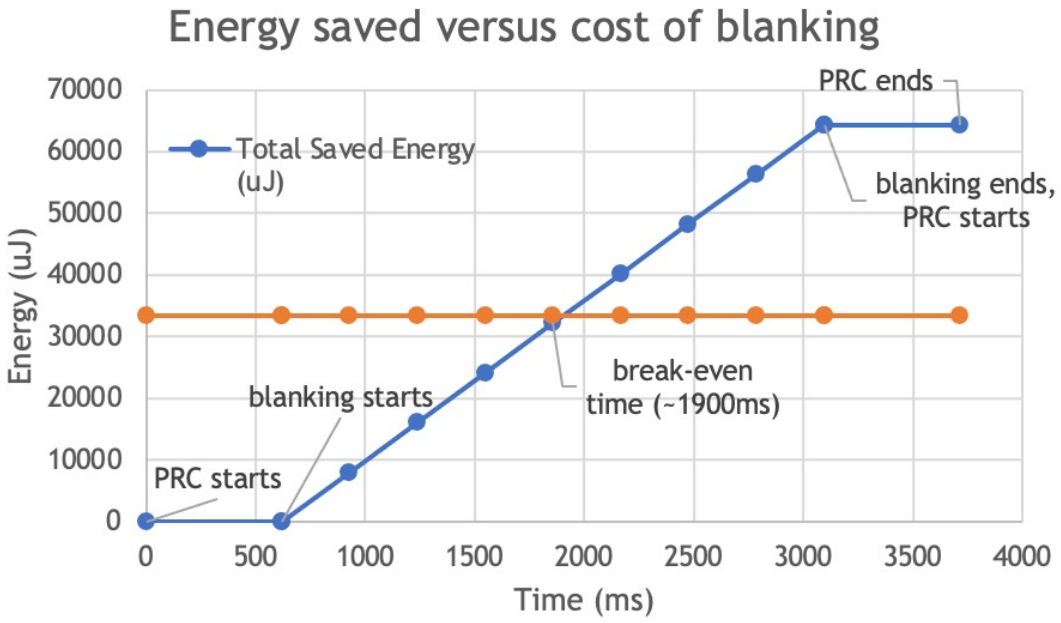
$$\frac{2 * P_{mission} * T_{PRC}}{(P_{idle} - P_{blanked})} \leq t_{blanked}$$



Partial Reconfiguration - Power

- Dynamic power drops from 27mW @ 50MHz to 1-2mW
- Static power drops from 246mW to 240mW
- Clock gating, in comparison:
  - Smaller dynamic reduction
  - Static unchanged

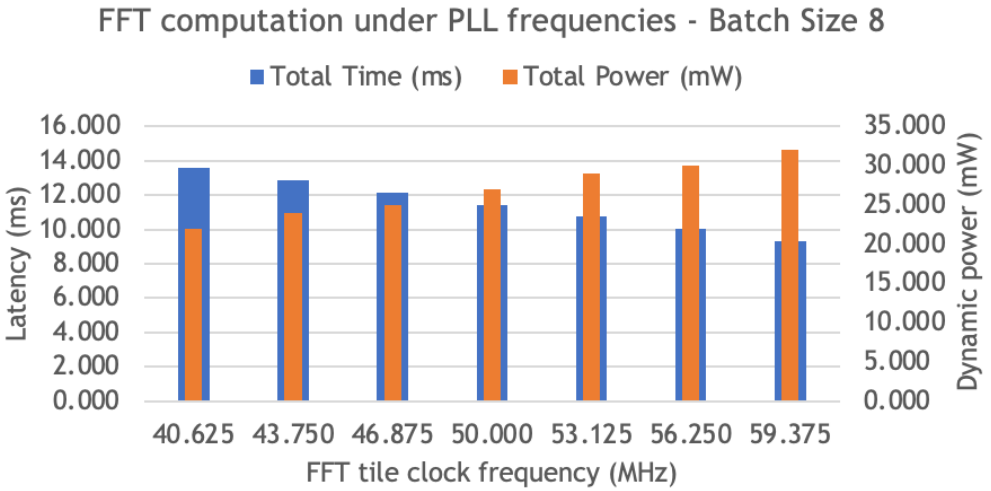
- Break-even for dynamic: ~1900ms

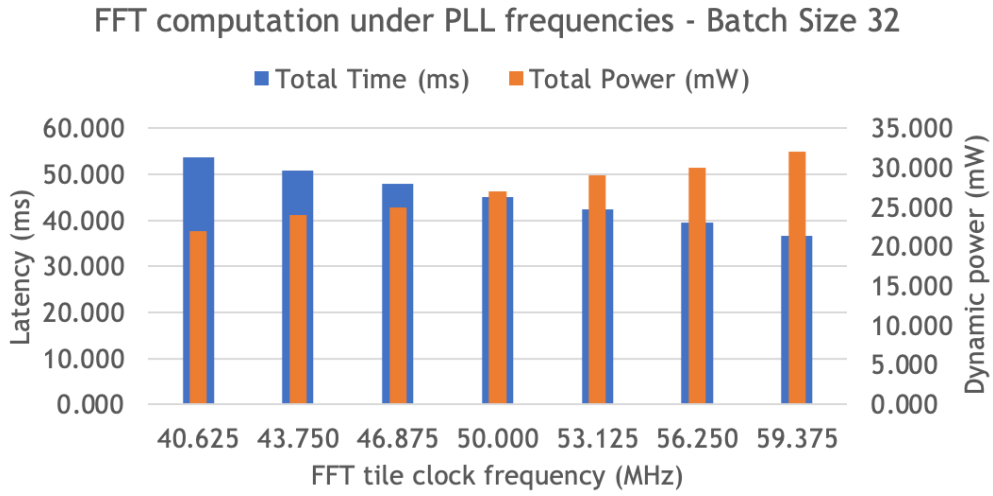


* This considers only dynamic power, but a total break-even time uses the same analysis

Cooperation

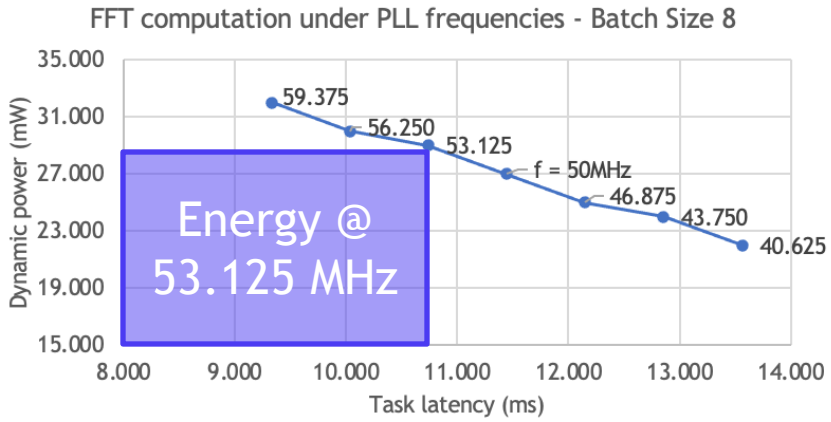
- Connect the two
  - implementation ongoing…
- Associate profile with current loaded bitstream
  - Power/reconfig. time profile from DPR flow
  - Latency profile from DFS

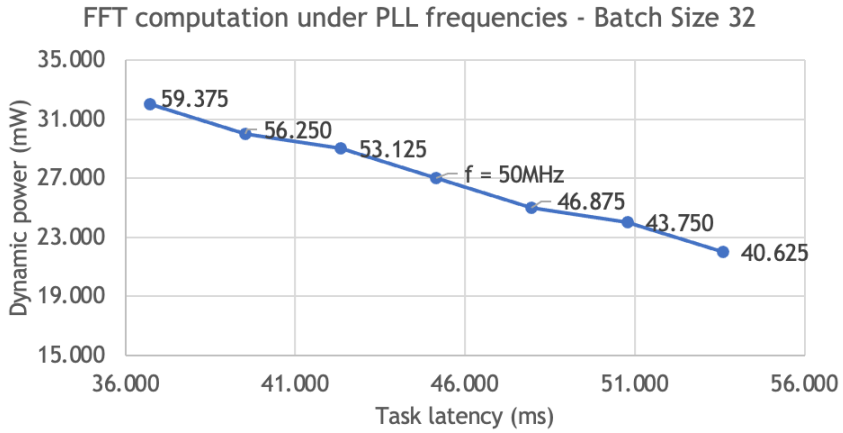




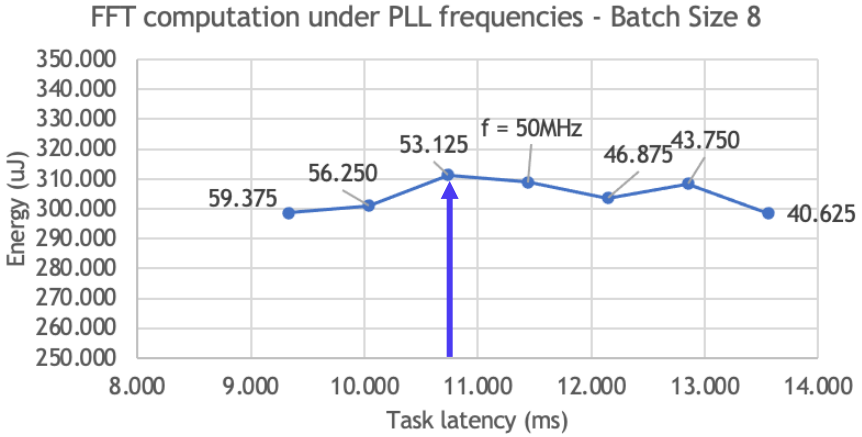
Cooperation results

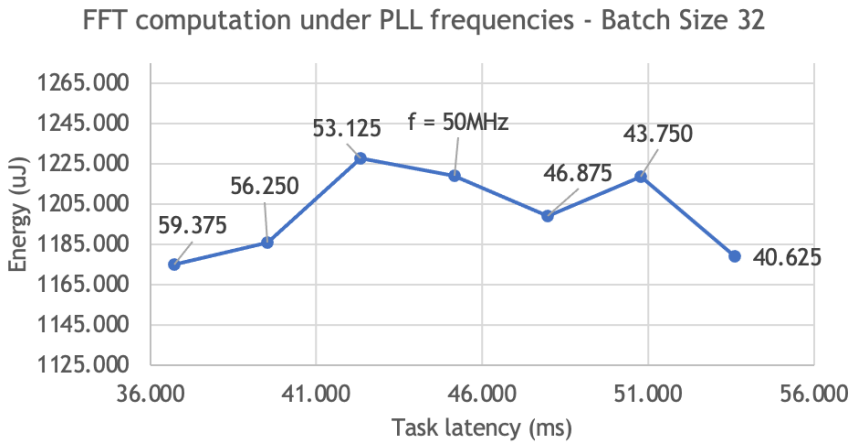
Power vs Time





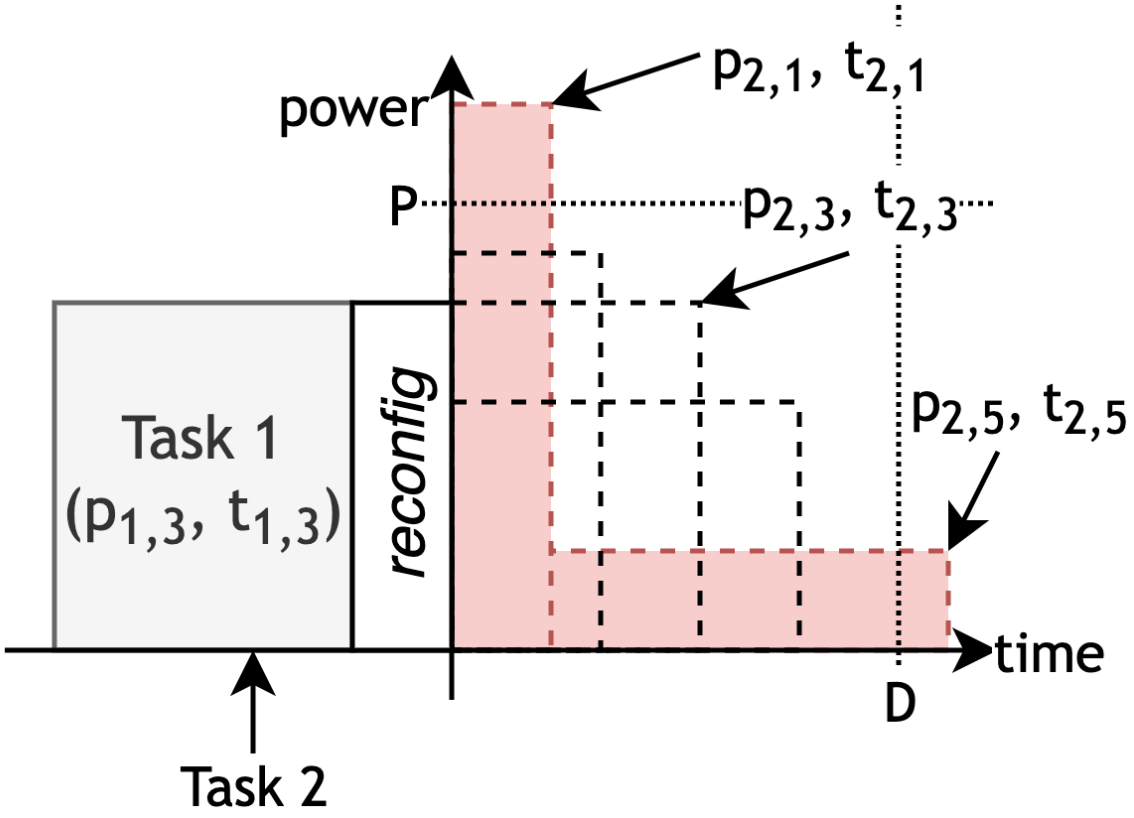
Energy vs Time





Power-aware scheduling

- Meet time, power, and energy constraints
- Flexibility to select a priority



Power-aware scheduling

Event	Cycle counter	Timestamp (ms)	Tile Frequency (MHz)	Dynamic power (mW)	Energy (uJ)
Start DPR	3937403448	196870.1724	50.000	27	61017.28785
Start DFS reconfig to select **53.125MHz**	3950234908	197511.7454	50.000	27	17322.471
Start work	3951856936	197592.8468	53.125	27	2189.7378
Start DPR	3996998813	199849.9407	53.125	29	65455.72165
Start DFS reconfig to select **56.250MHz**	4009830273	200491.5137	53.125	29	18605.617
Start work	4011451031	200572.5516	56.250	29	2350.0991
Start DPR	4056536866	202826.8433	56.250	30	67628.7525

Table derived from software printout

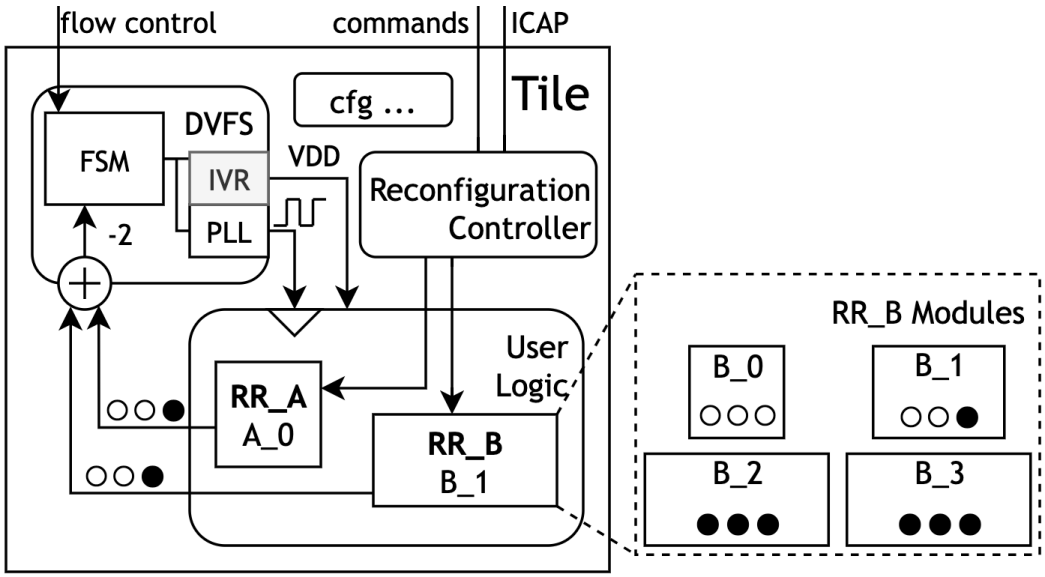
Characterization of DPR time and power provided from DPR flow

Comparison

	No DFS/DPR (existing)	DFS (complete)	DPR (up to date)	DFS+DPR (to complete)
Power	Fixed	Flexible based on frequency	Flexible based on bitstream	Most flexible
Area	Smallest	Extra for PLL	Padded to fit pblock	Largest (PLL not in pblock)
Latency	Fixed	Can go faster or slower	Fixed, more with reconfig.	Flexible
P&R time	Slower	Slower	Faster	Faster

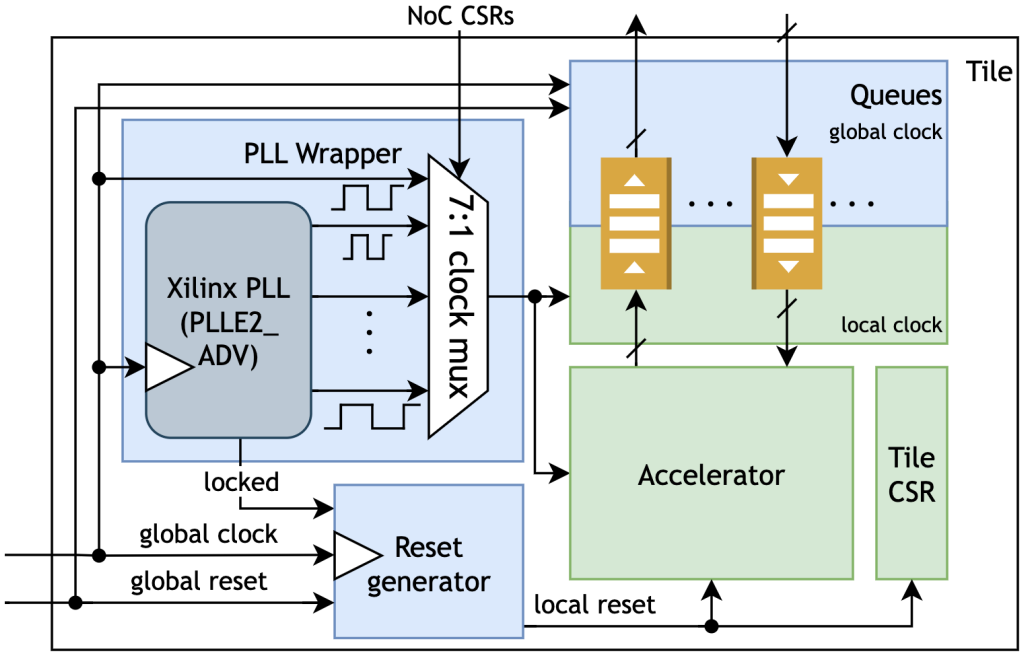
Future work

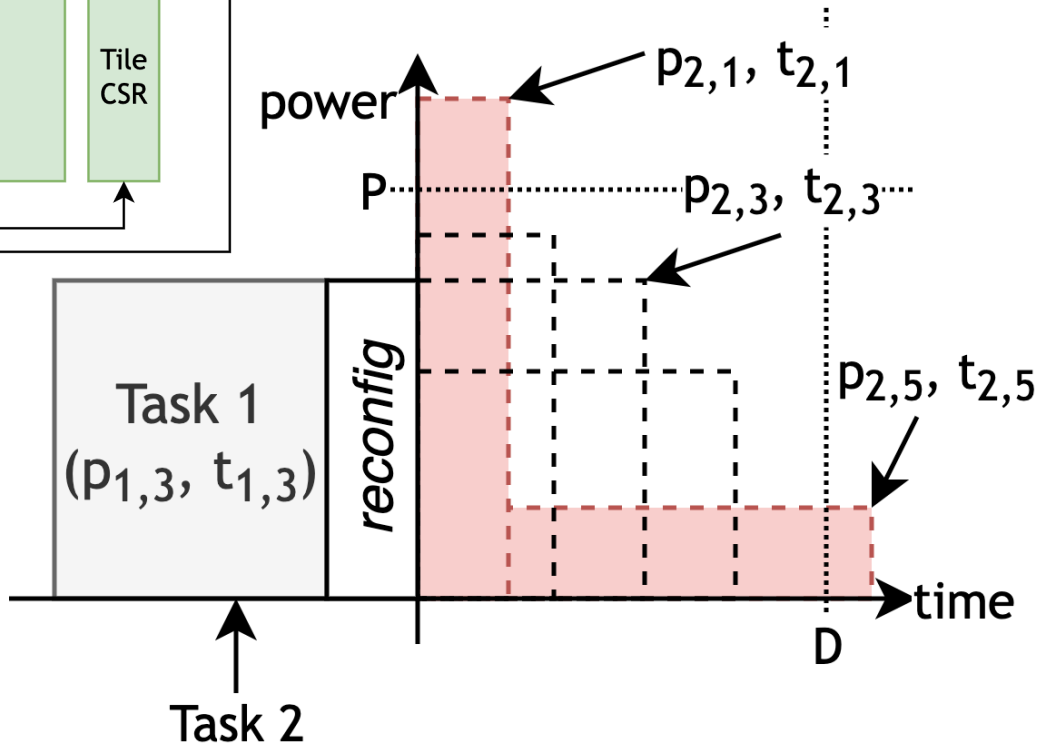
- This project: infrastructure/ connection



- Next steps: finer-granularity
  - Autonomy for better response time
  - Sub-tile regions offset system-derived freq.
  - Measure real-time power during execution

Thank you!





COLUMBIA | ENGINEERING
The Fu Foundation School of Engineering and Applied Science

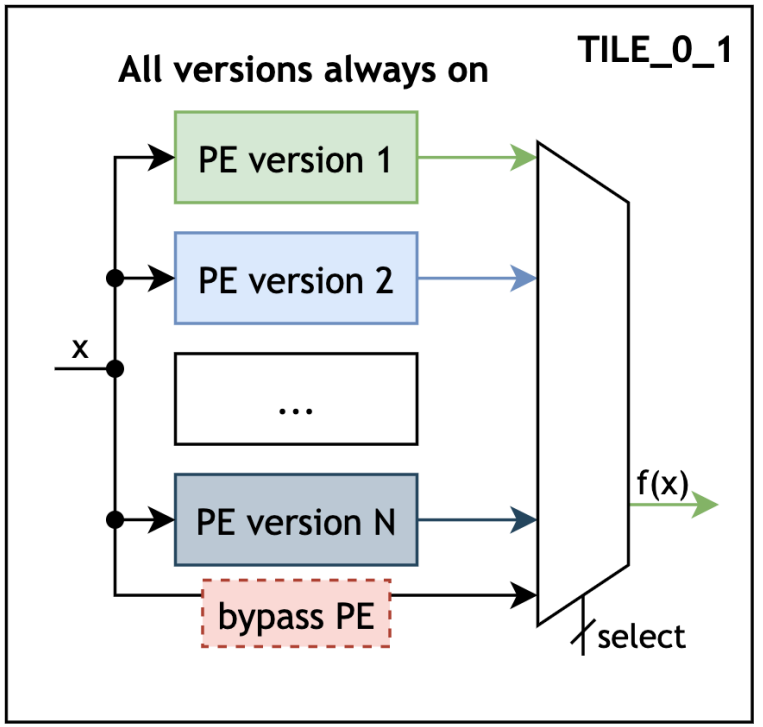
Backup slides – nested regions

Purpose

- Heterogeneous workloads ⇒ heterogeneous activity in an SoC
- Tile-level adaptation in SoCs is proven
- Sub-tile regions allow for more aggressive control and power savings

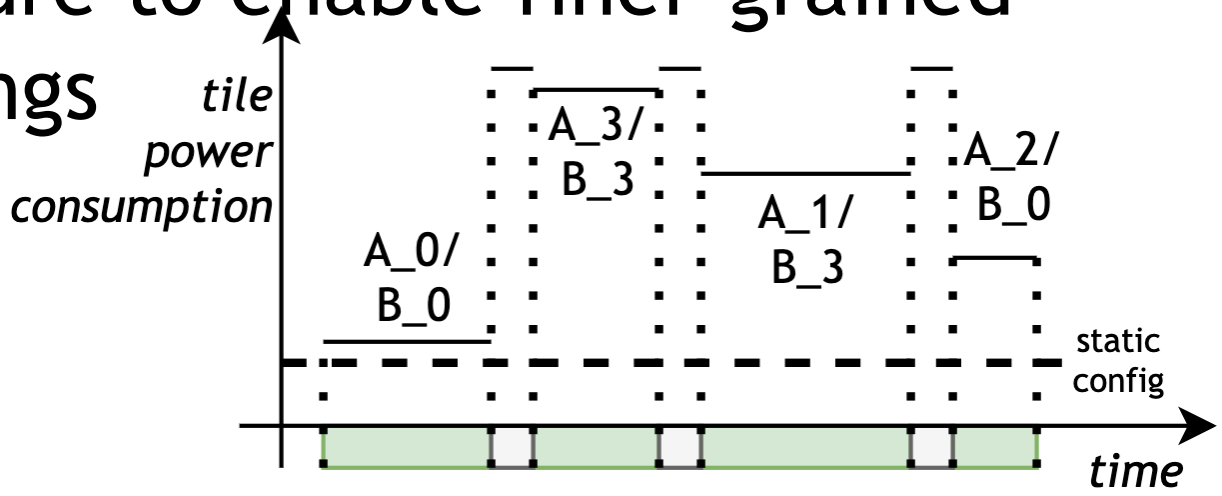
- Heterogeneous workloads ⇒ heterogeneous activity in an SoC
- Tile-level adaptation in SoCs is proven
- Sub-tile regions allow for more aggressive control and power savings

(Mantovani et al., 2016)



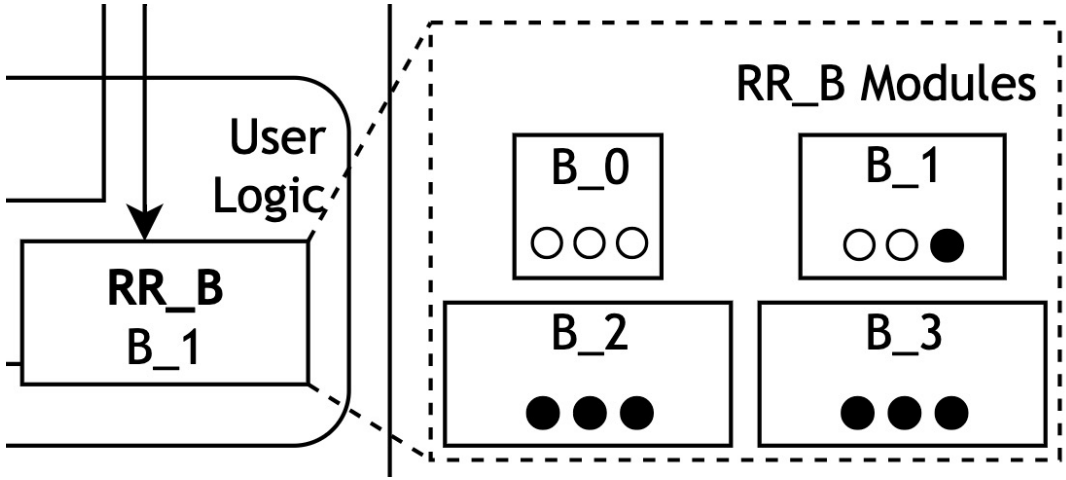
Solution – FPGA-based SoCs

- Dynamic partial reconfiguration (DPR)
  - Reprogram specific regions of the FPGA
  - Implemented for tiles in ESP
  - Extend using nested partial reconfiguration
- **Strength**: connect to system-level infrastructure to enable finer-grained power savings



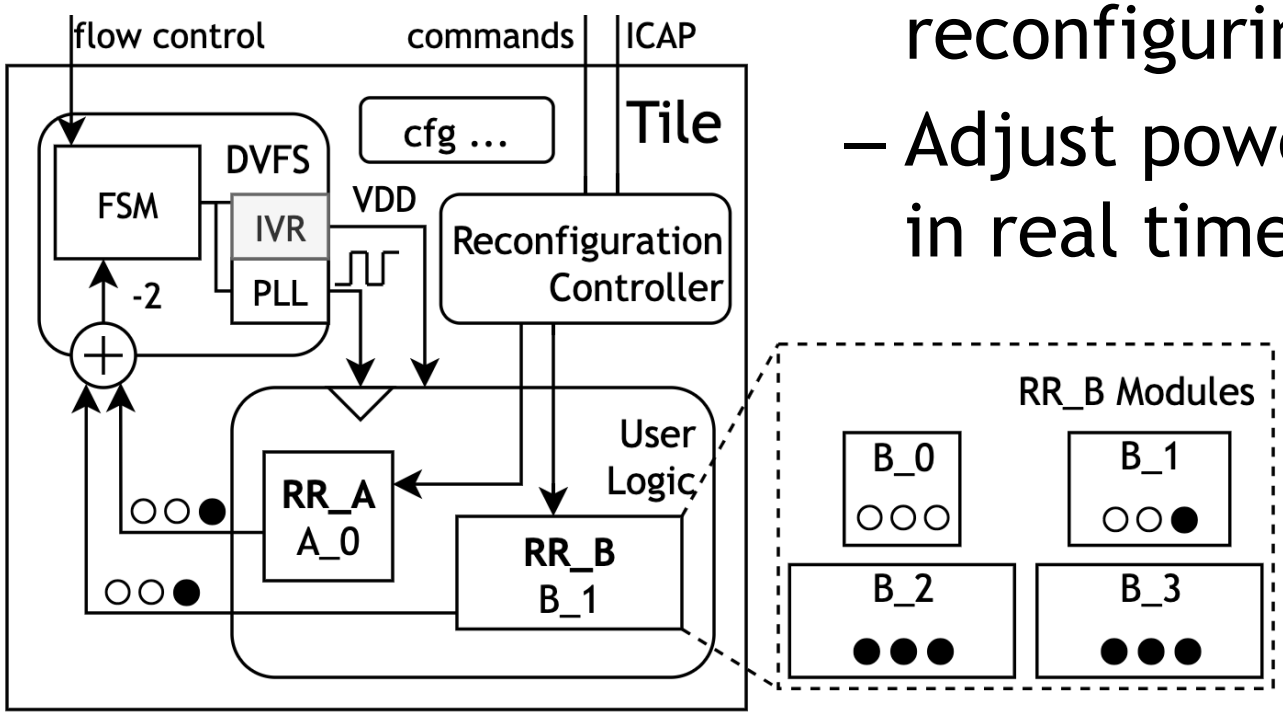
System design flow

- Existing flow: tile must contain resources for each tile configuration
- New constraints
  - Specify nested regions
  - Consider resource usage
    - CLB
    - DSP
    - BRAM
    - FF



Hardware

- Update PLL
- Wrapper/controller
  - Isolate when reconfiguring/blanked
  - Adjust power allocation in real time (DFS)

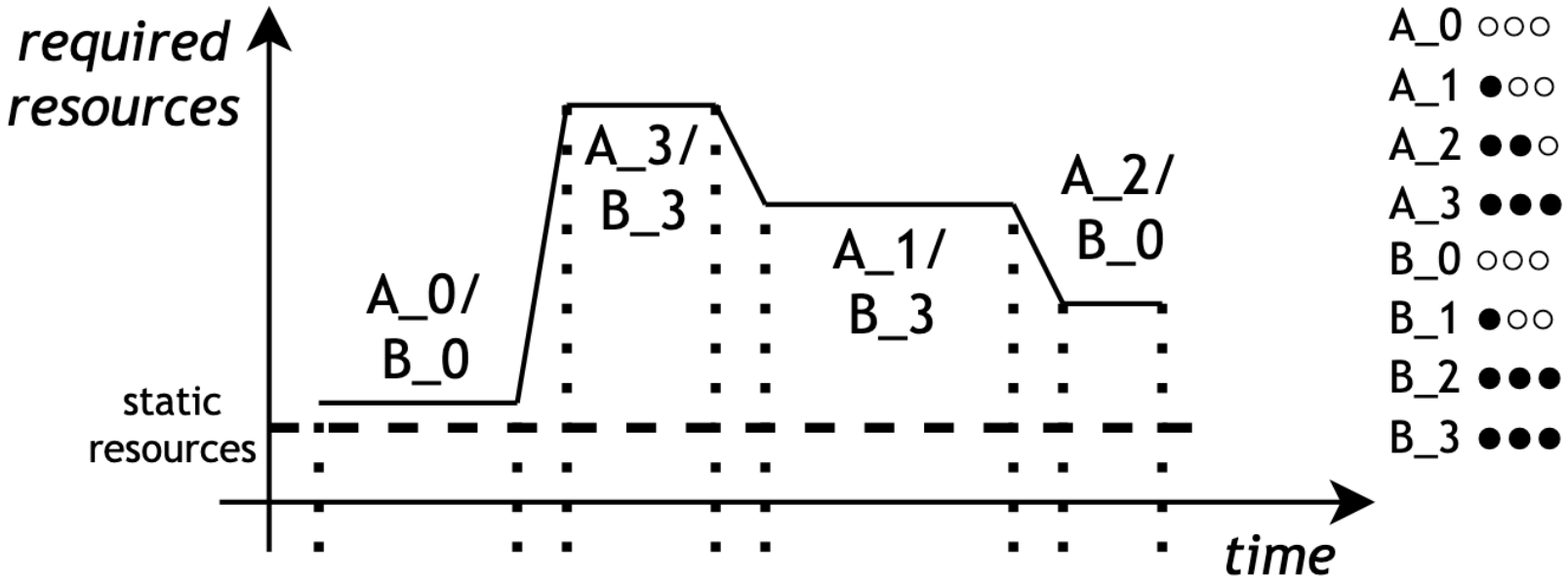


Software

- Bare-metal drivers
- Request validation
  - Cannot load nested regions if their parent configuration is not loaded

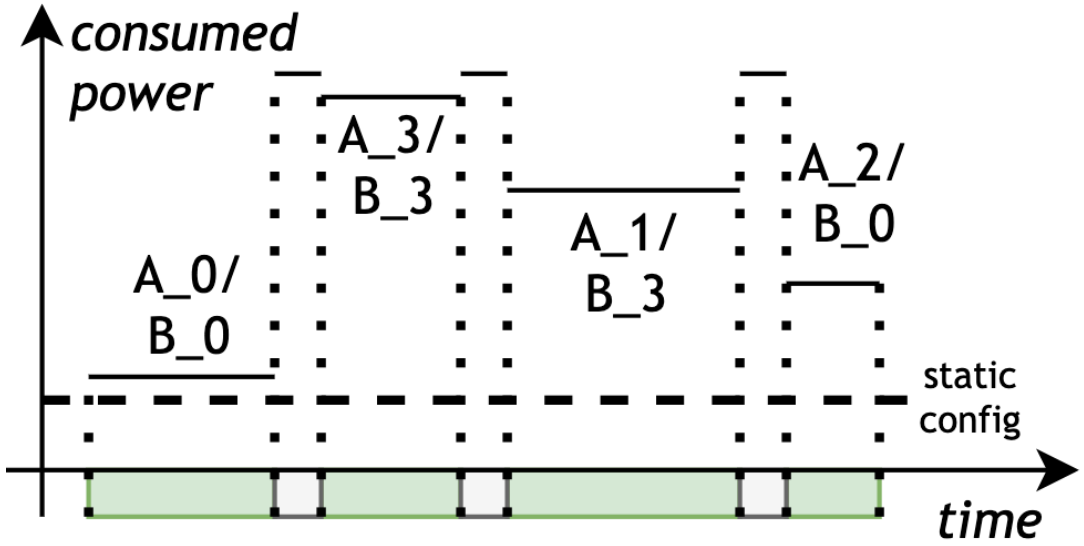
Expected Results – Power

- Measure power/energy consumed
  - During FPGA reconfiguration
  - Under different configurations of a tile



Expected Results – Power

- Evaluate reconfiguration cost
  - Stalling time
  - Power of using ICAP



- Compare approaches (power, area, time)
  - DFS (Mantovani et al., 2016)
  - Blanking bitstreams for power gating
  - Swappable tiles (Seyoum et al., 2023)
  - **Sub-tile reconfiguration** (my approach)

Expected Results - Accelerators

- Design composability for a reconfigurable pipeline
- Then adapt applications to an accelerator