Physical design implementation of on-chip segmented bus to reduce communication energy

Jin Guo

Antonis Papanikolaou

Pol Marchal

Francky Catthoor

OUTLINE

- **Introduction**

- **Two steps of network energy optimization**
  - **-- Block ordering**
  - **-- Activity aware floorplanning**

- **Conclusions**

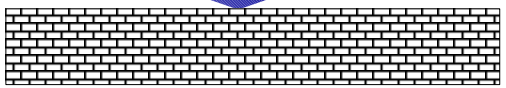
Low-energy operation should be targeted in all the design phases



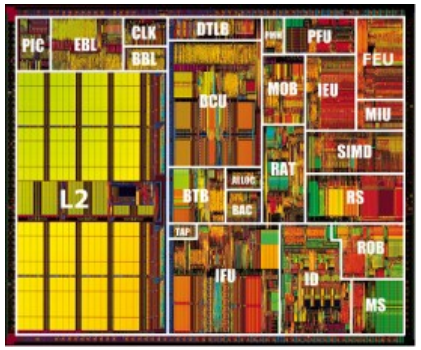

Battery Power driven

System-level design

Links between different design phases

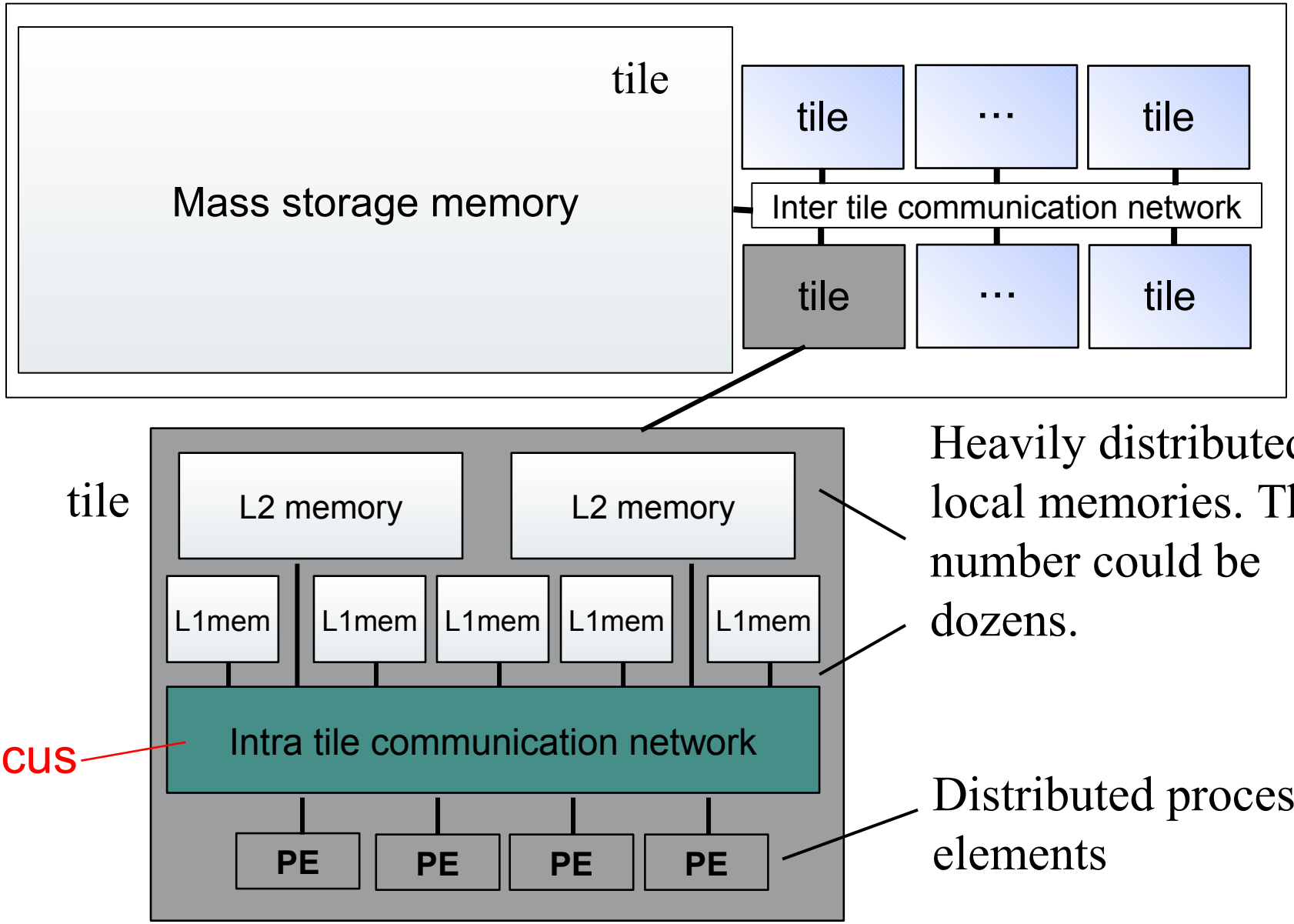


Physical design

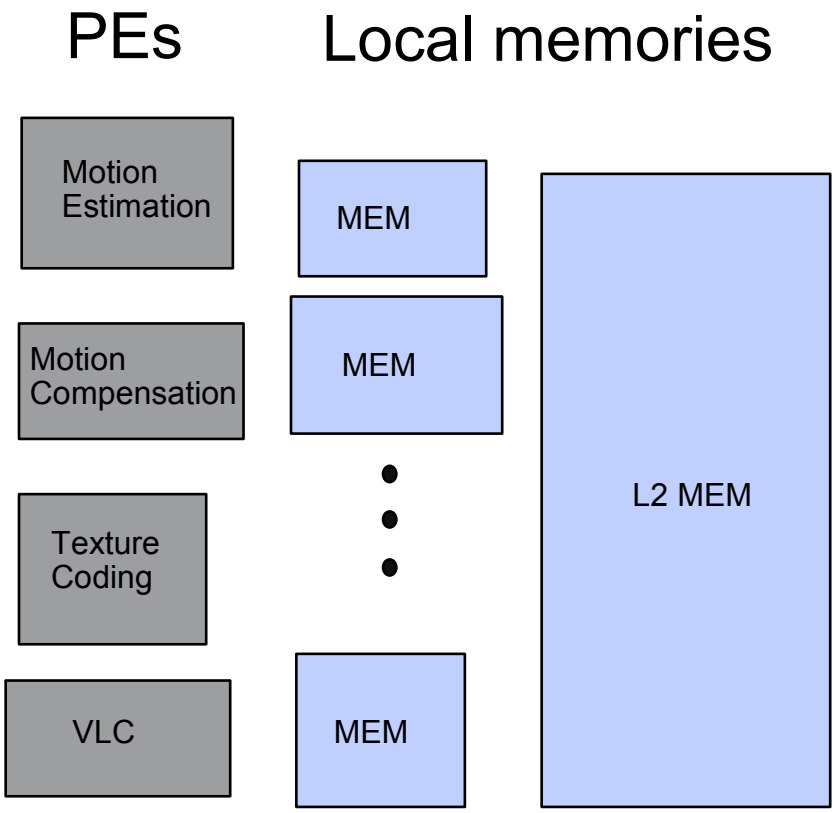


Energy efficient operation is a key issue for embedded systems

We focus on the intra-tile communication networks



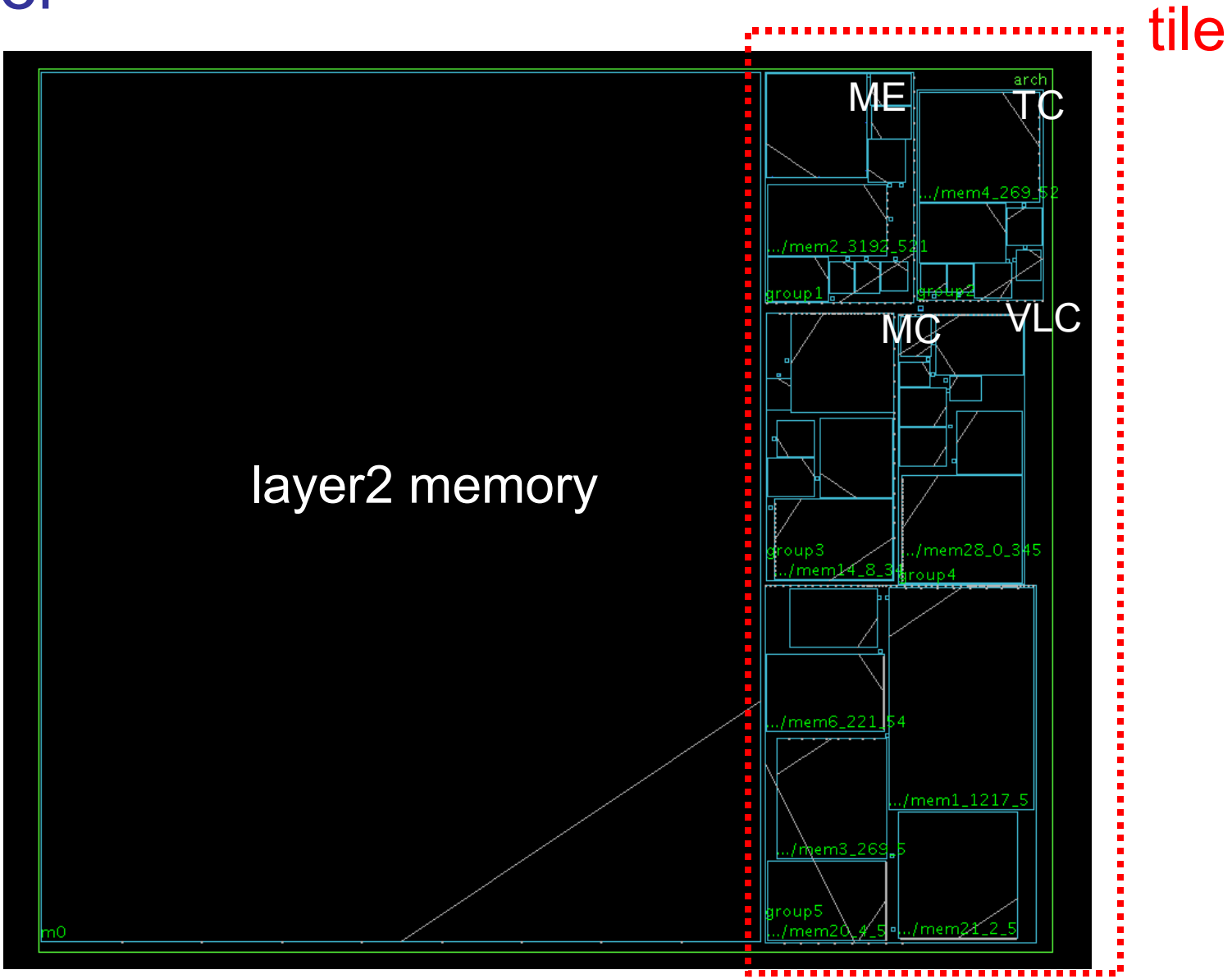
Hard macro block based floorplanning



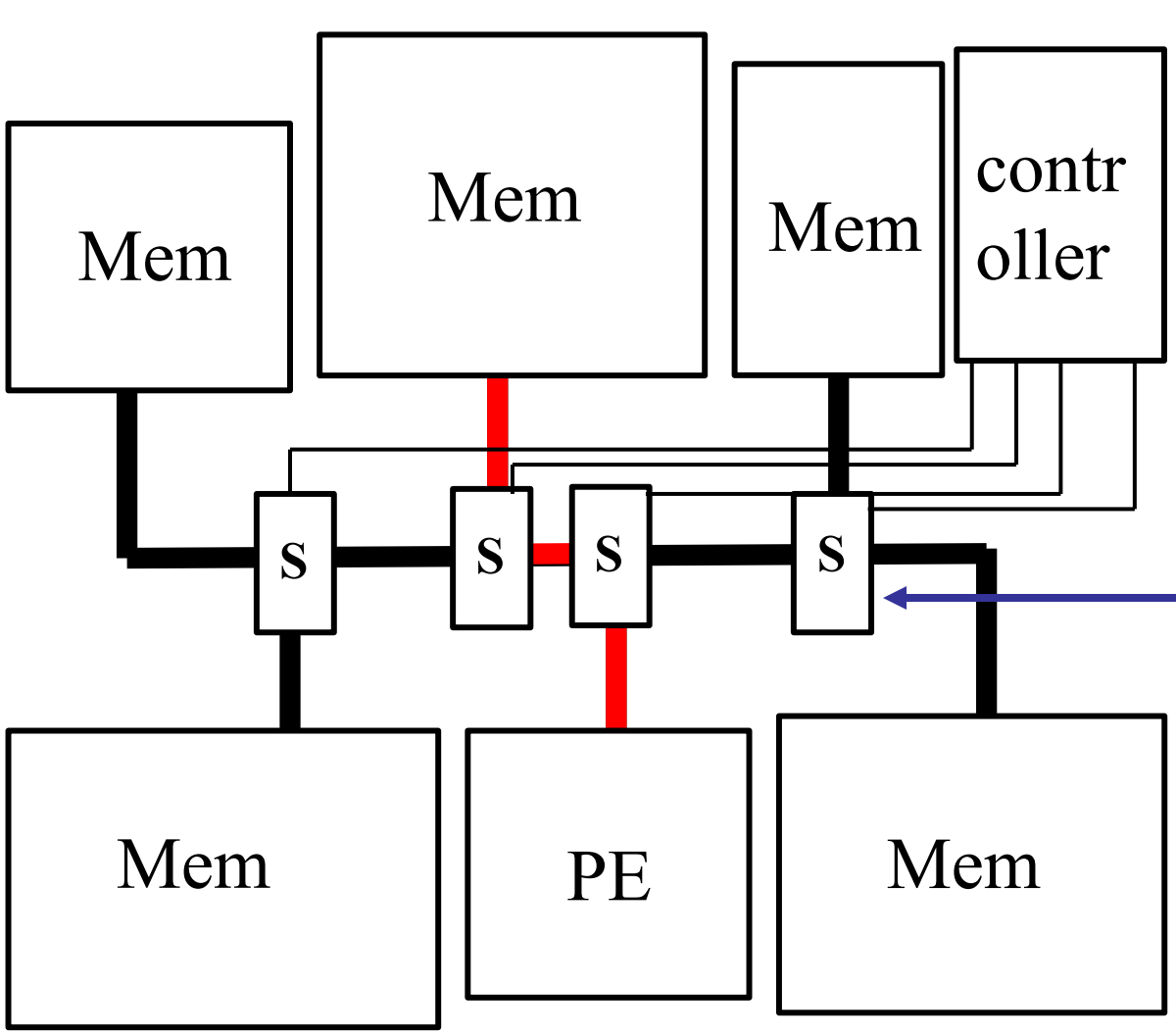
MPEG4 encoder

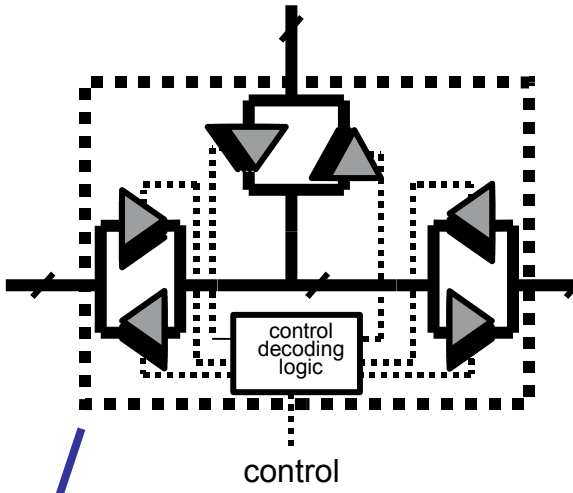
- The floorplanning is at block level-> IP reuse
- Size & shape of the macro block are fixed -> avoid overruling the already made decisions at higher level

Floorplan result demonstrated on MPEG4 encoder



On-chip segmented bus architecture based on shared bus

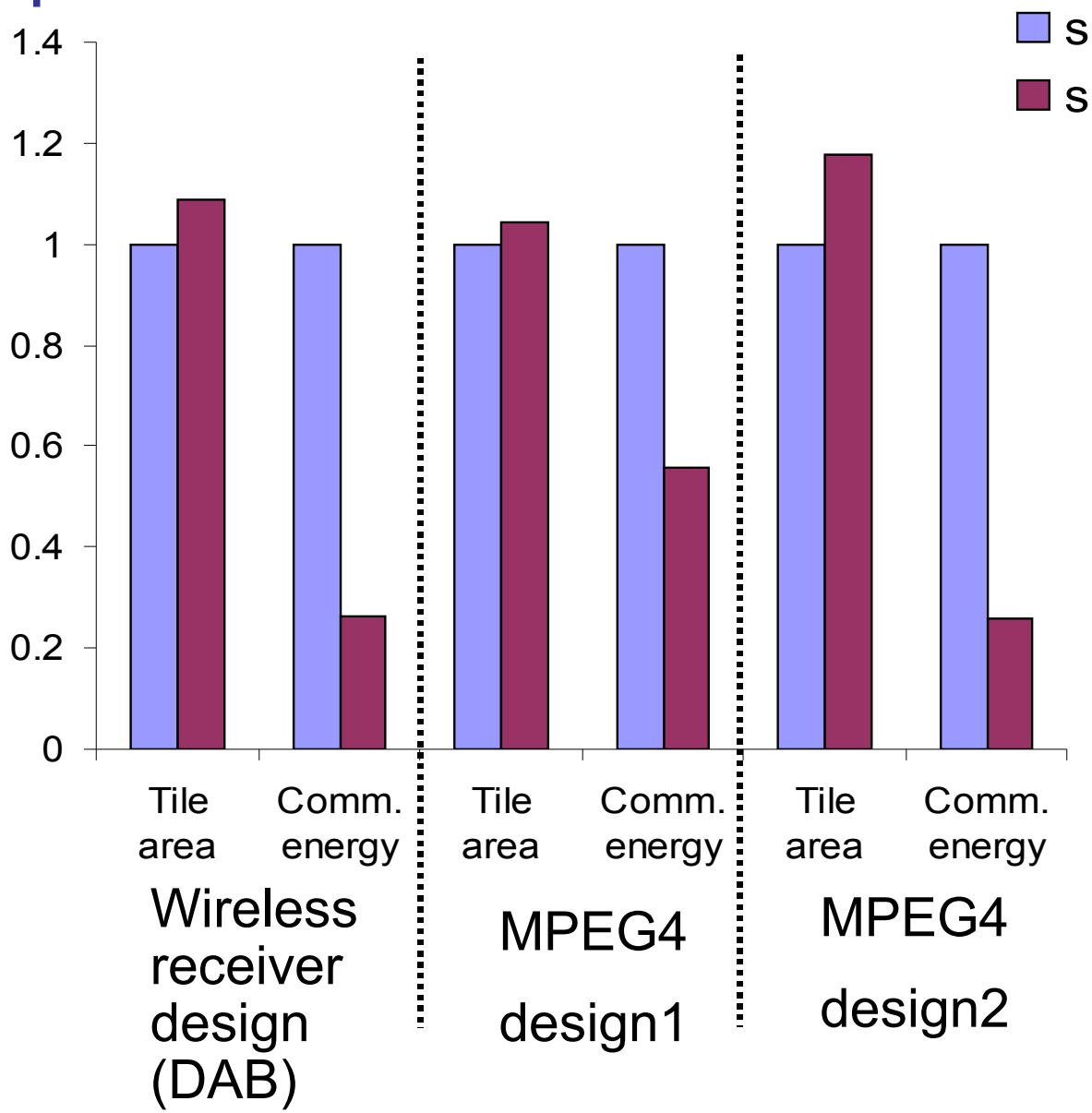




Switches

Shared bus architecture

Segmented bus vs. shared bus for different application drivers



Segmented bus reduces the wire load significant, hence reduces the network energy consumption and the critical path delay

OUTLINE

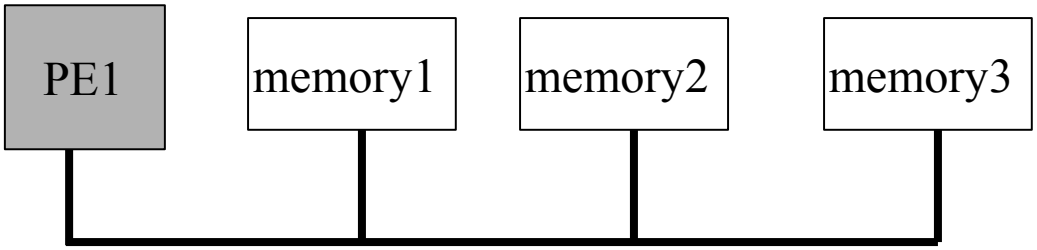
- Introduction

- **Two steps of network energy optimization**
  -- Block ordering
  -- Activity aware floorplanning

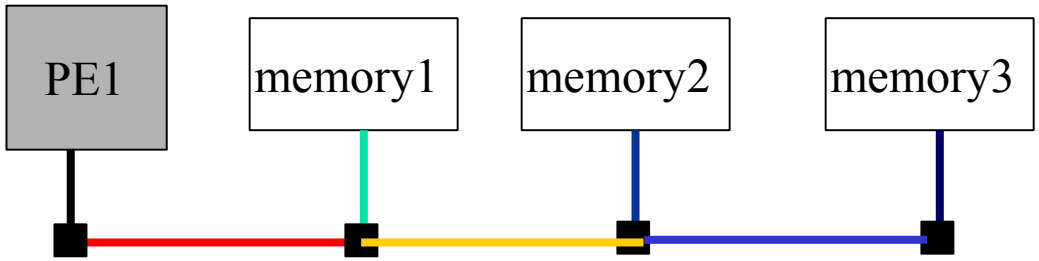
- Conclusions

Segmented buses bring new challenges --block ordering (Block-level netlist)

- Shared bus: one single net connects all the blocks, there is no variations for the netlist

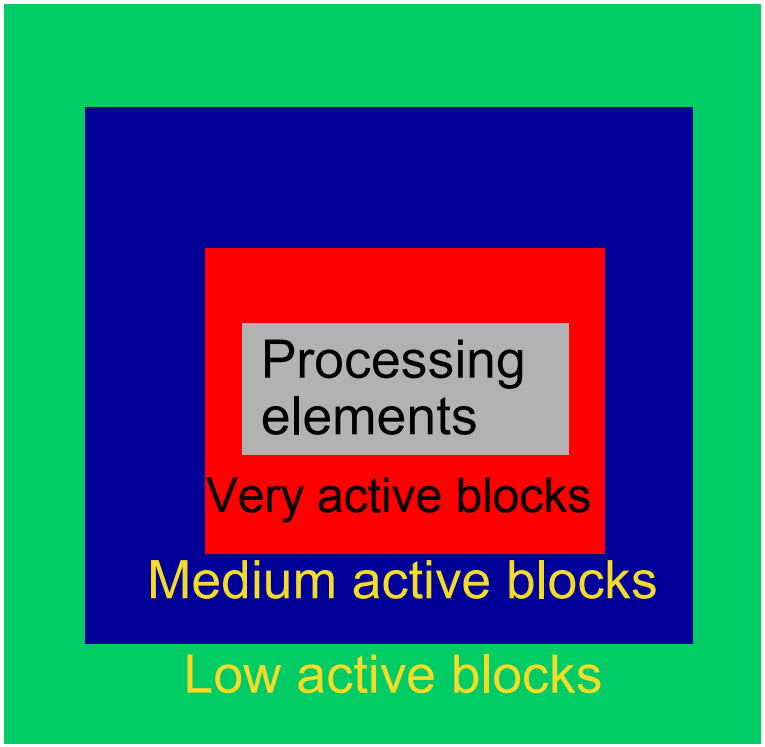


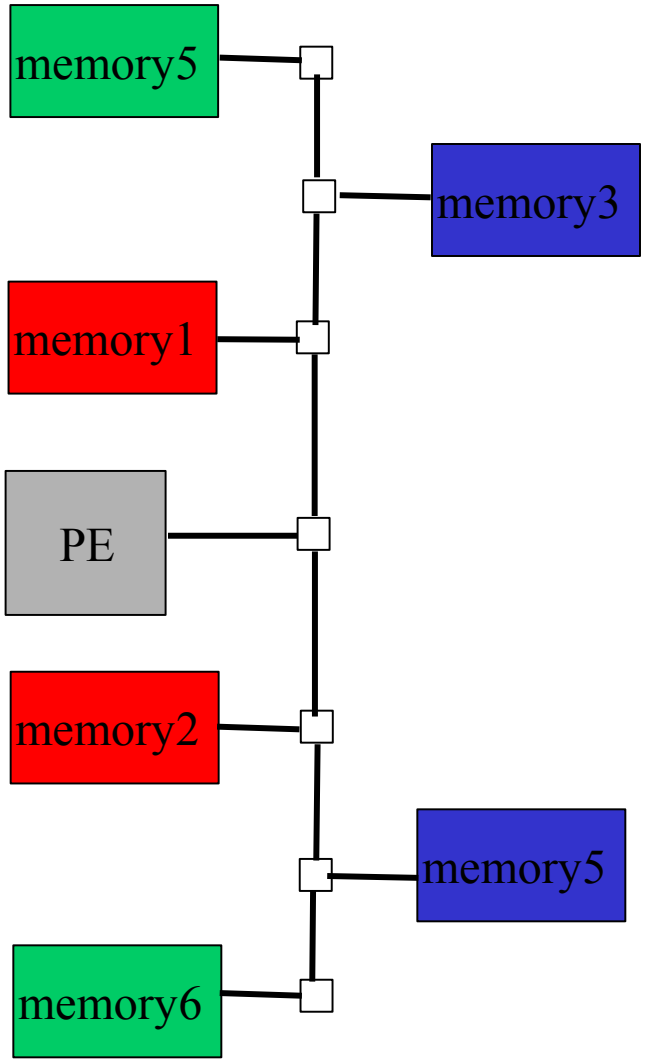
- Segmented bus: The net is divided into segments by the switches. The activity of the segment is determined by the netlist



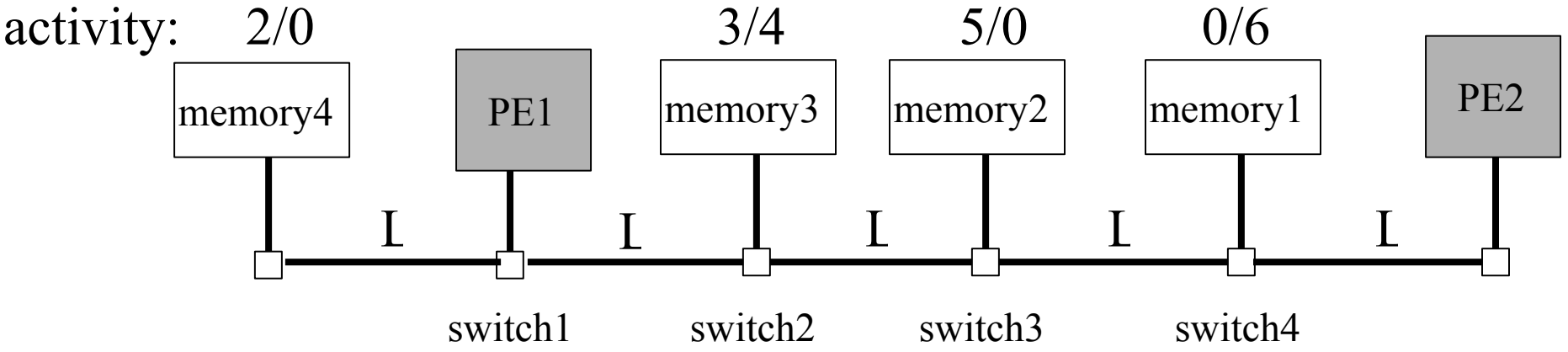
Activity-aware netlist for energy optimal solutions

Blocks that communicate very frequently should be ordered close to each other and vice versa



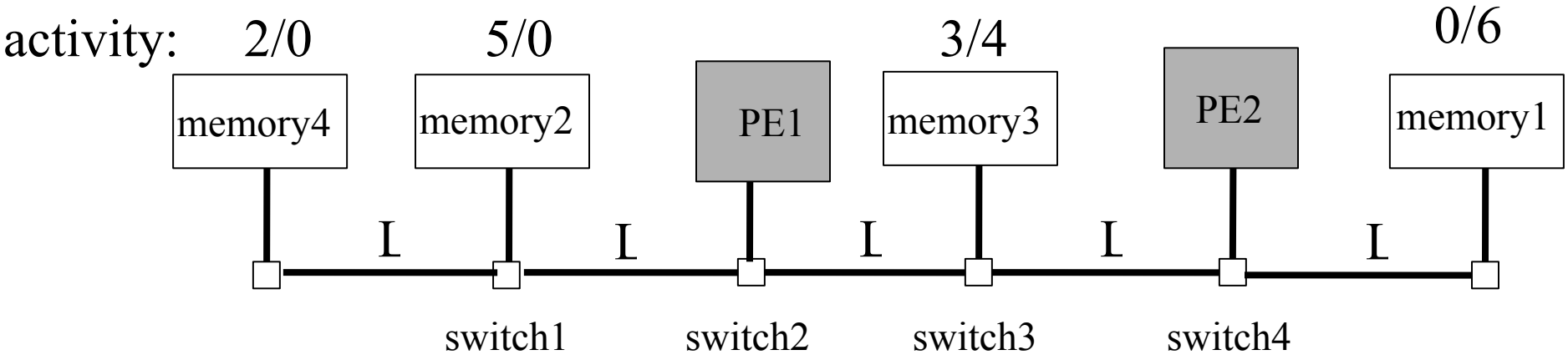


Two steps optimization for segmented bus energy consumption--Step1. Block ordering



a. Non optimal block ordering decision for energy

Cost=2*L+3*L+4*3L+5*2L+6*L=33L

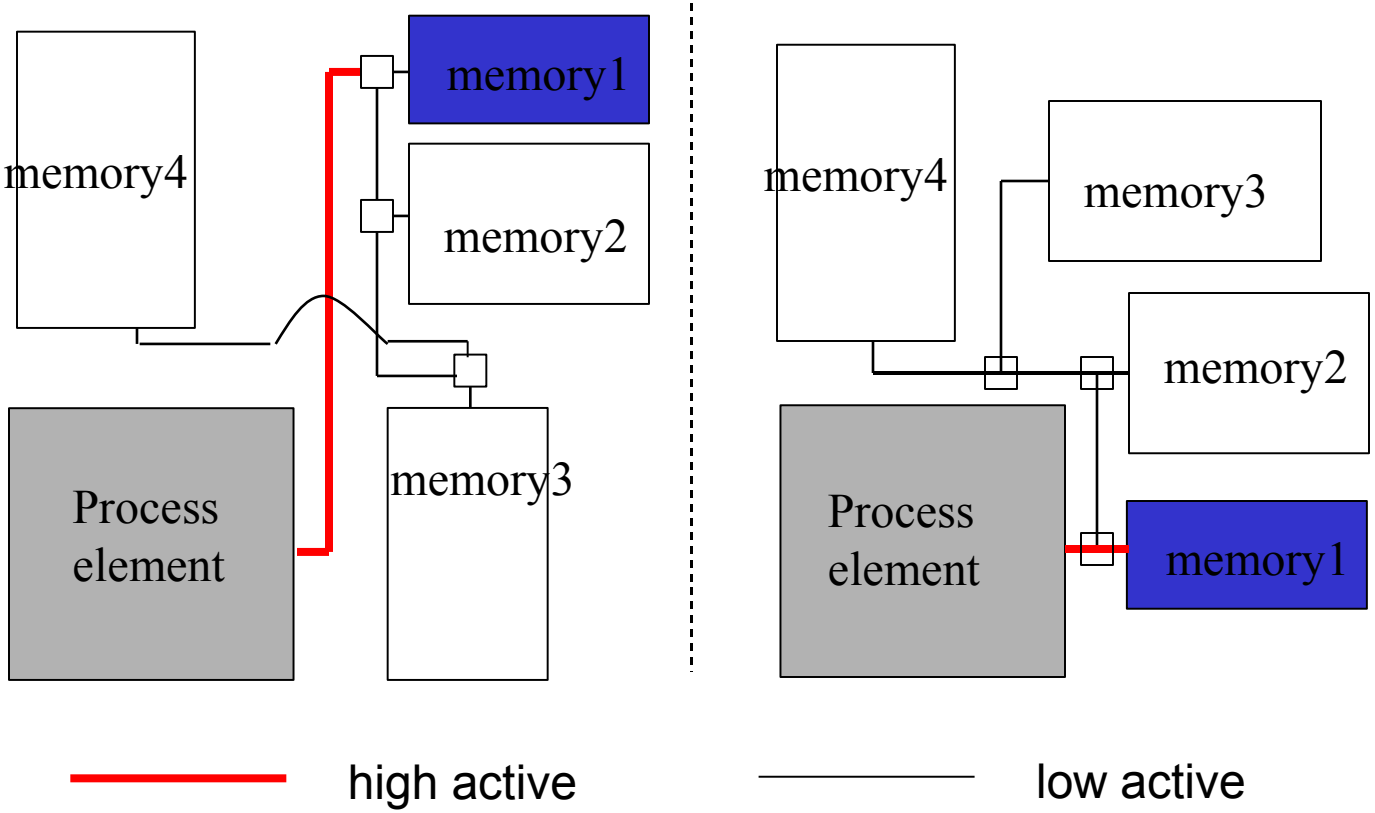


b. Optimal block ordering decision for energy

Cost=2*2L+5*L+3*L+4*L+6*L=22L about 33% cost saved

Two steps optimization for segmented bus energy consumption--Step2. Activity aware floorplanning

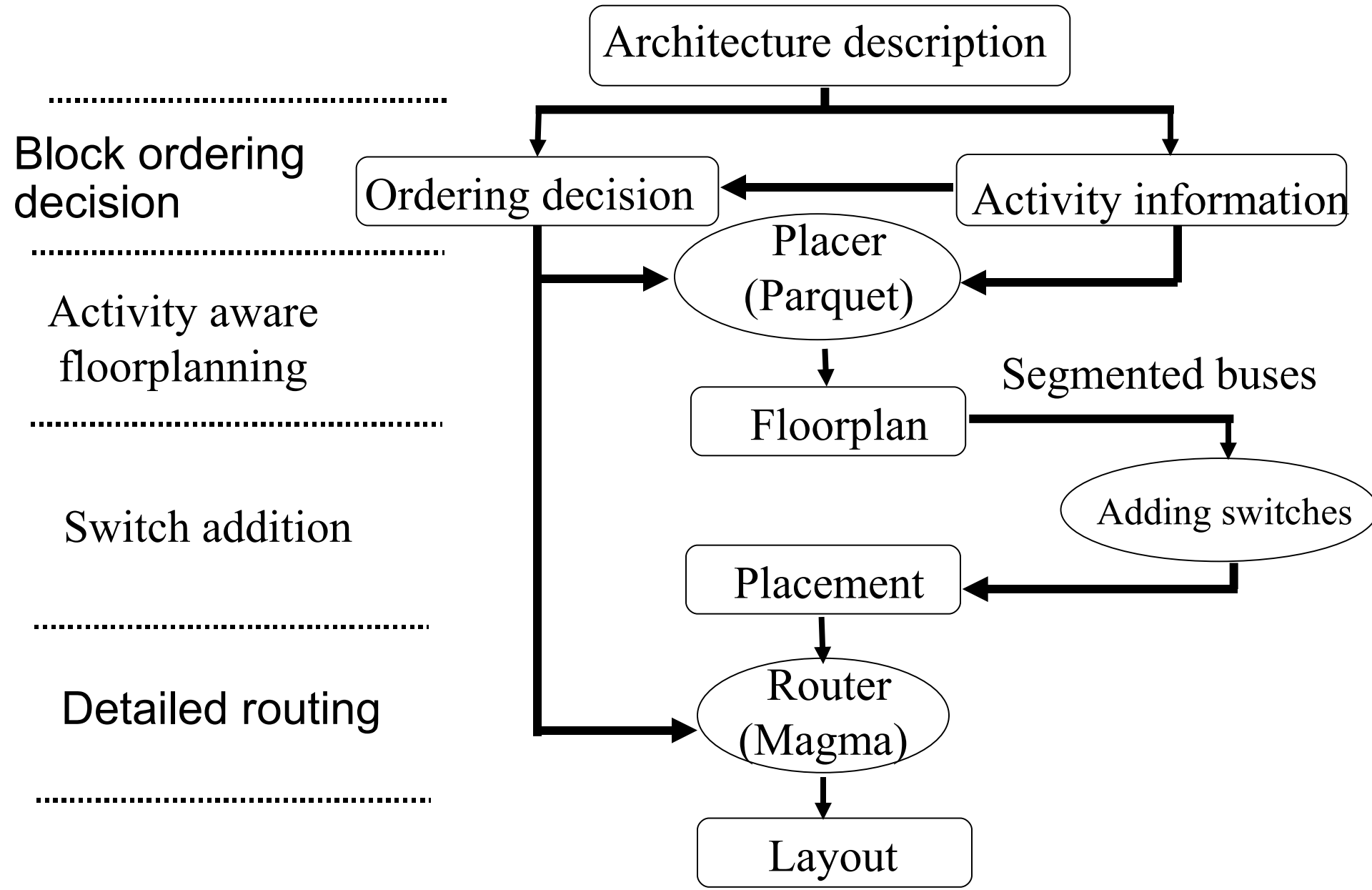
-- minimizing the physical length for the high active segments



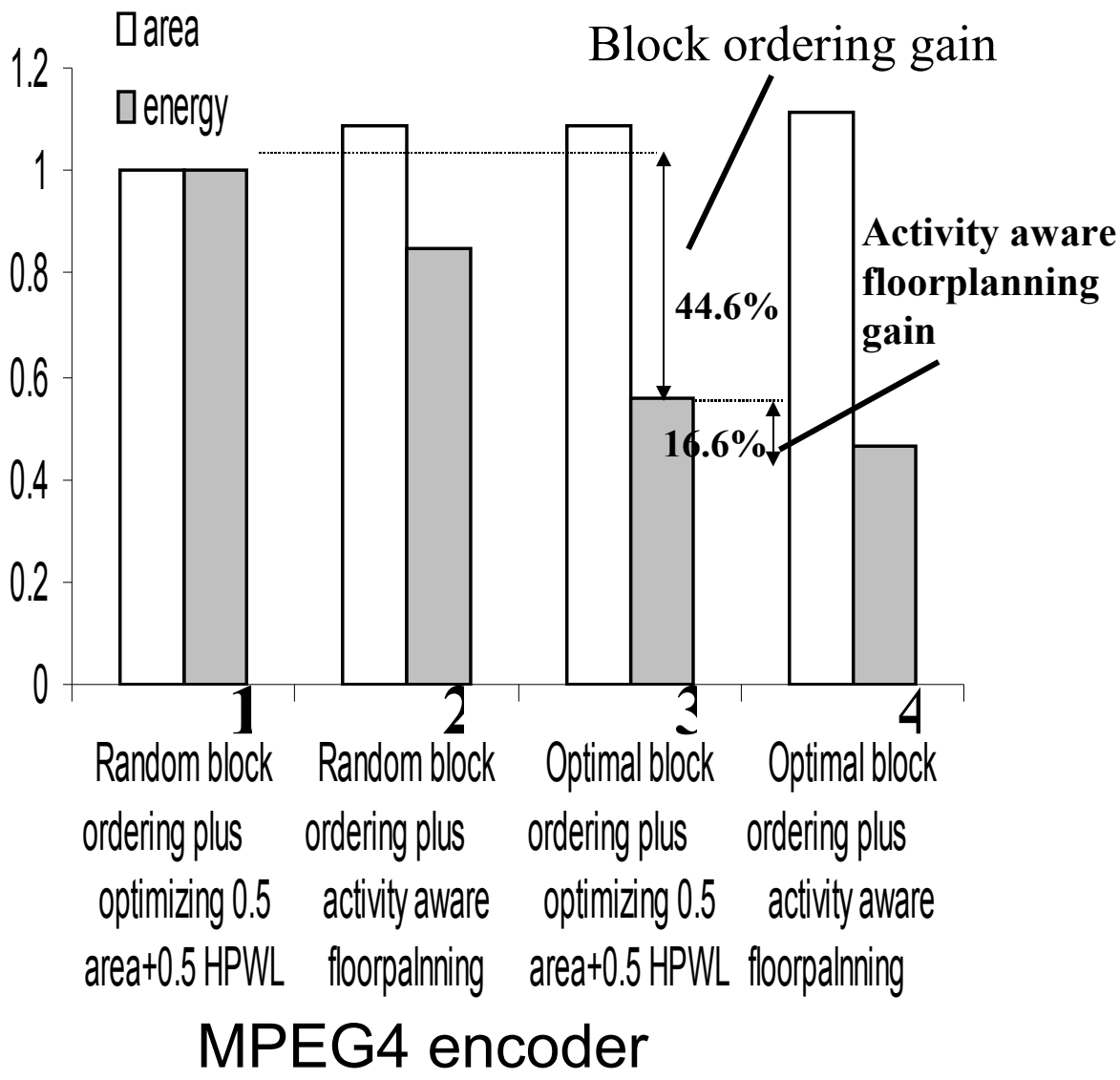
a. Segmented buses with non activity aware floorplanning

b. Segmented buses with activity aware floorplanning

Energy efficient methodology flow-graph



Impact of block ordering and activity aware floorplanning—linear topology



-- The network energy cost is more sensitive to block ordering than to activity aware floorplanning

-- If the two steps in terms of block ordering and activity aware floorplanning are coupled efficiently, energy gains of a factor of 2.16 can be achieved

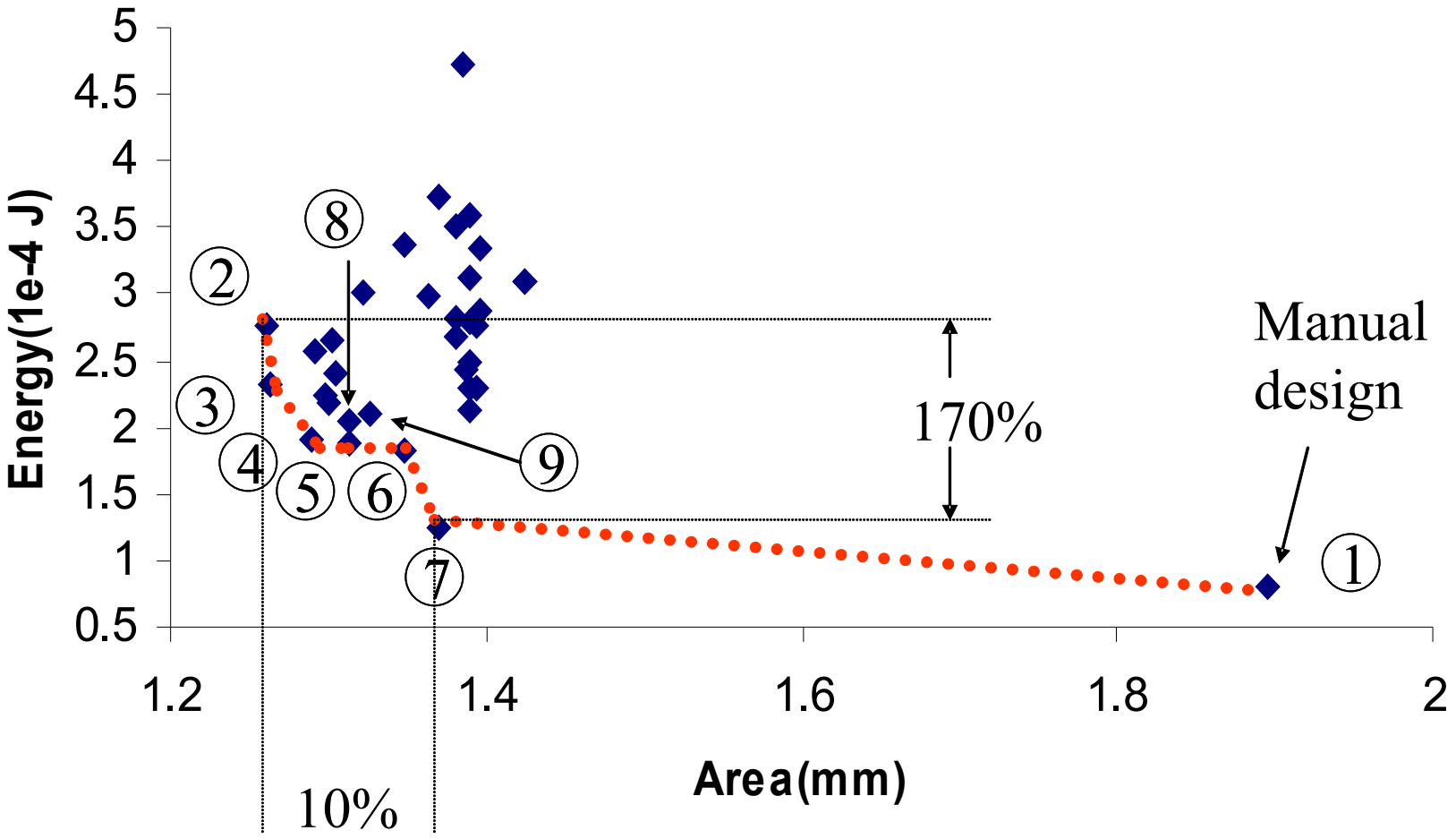
Conclusions

- We can significantly improve network energy consumption with a slight penalty in total chip area using the segmented bus.

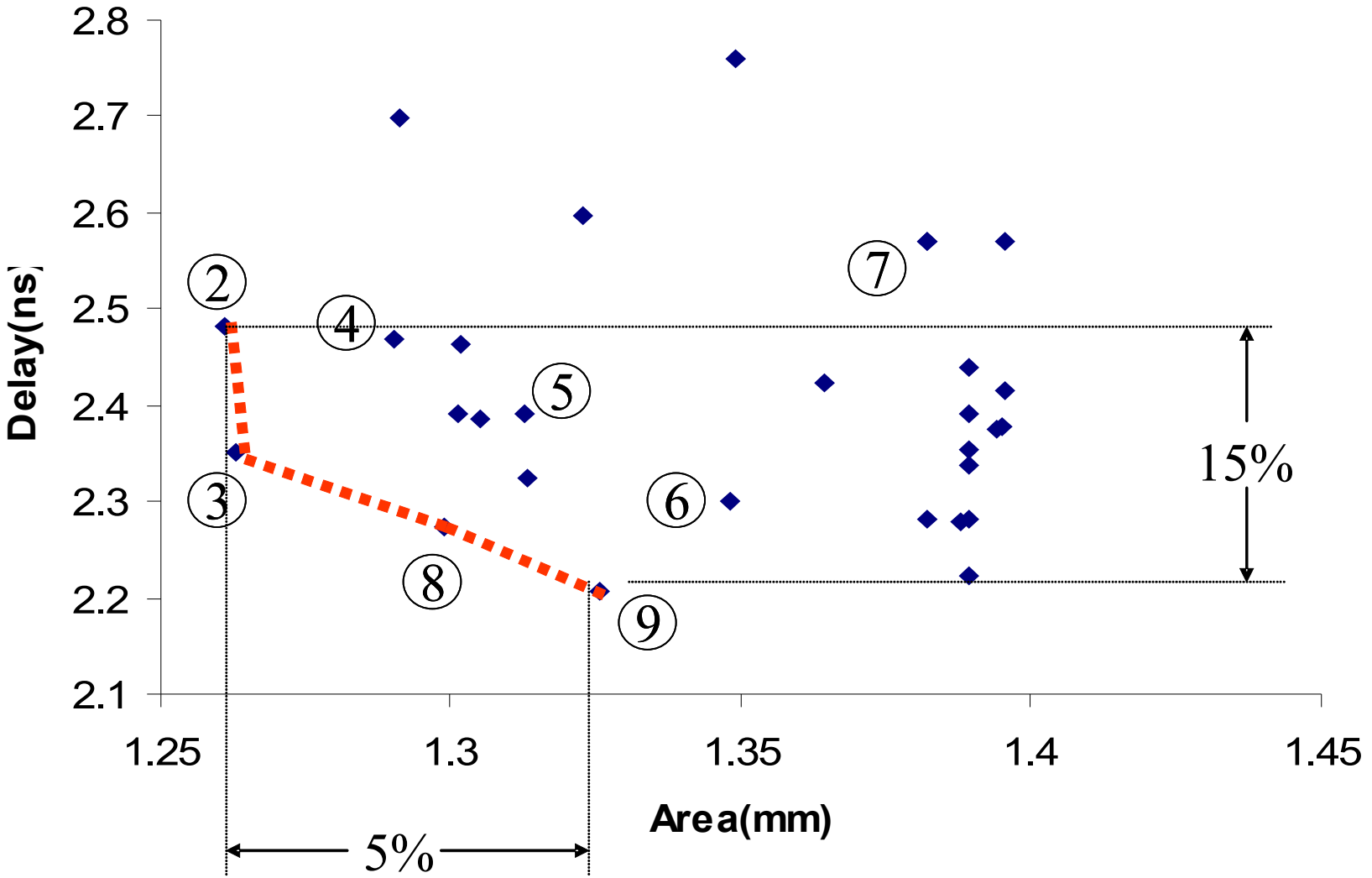
- Energy optimization should be targeted in all phases of the design

- The communication energy is more sensitive to the block level netlist topology compared to the floorplanning for segmented communication architecture

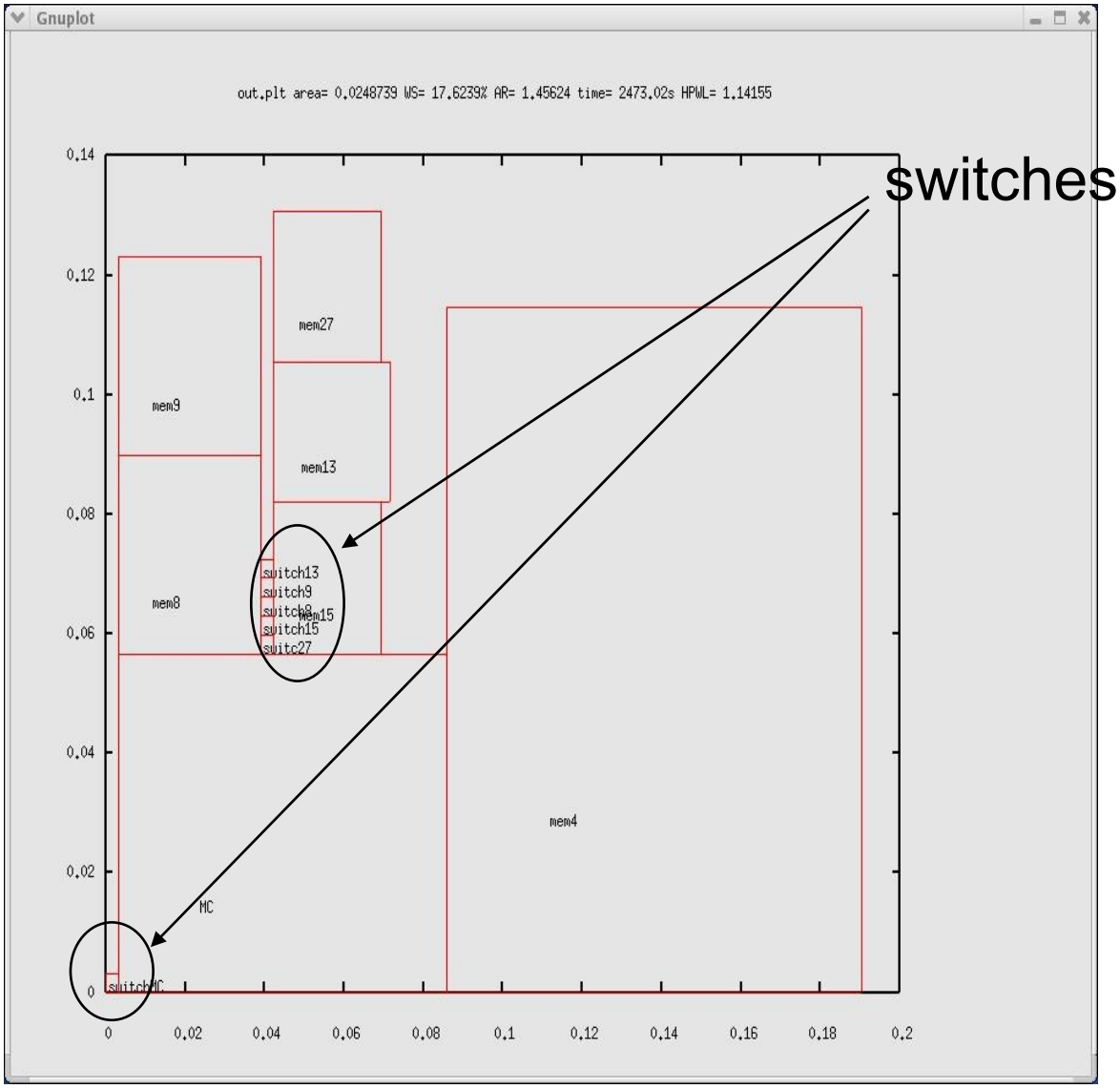
Area/Energy Pareto curve for MPEG4 encoder



Area/Delay Pareto curve for MPEG4 encoder

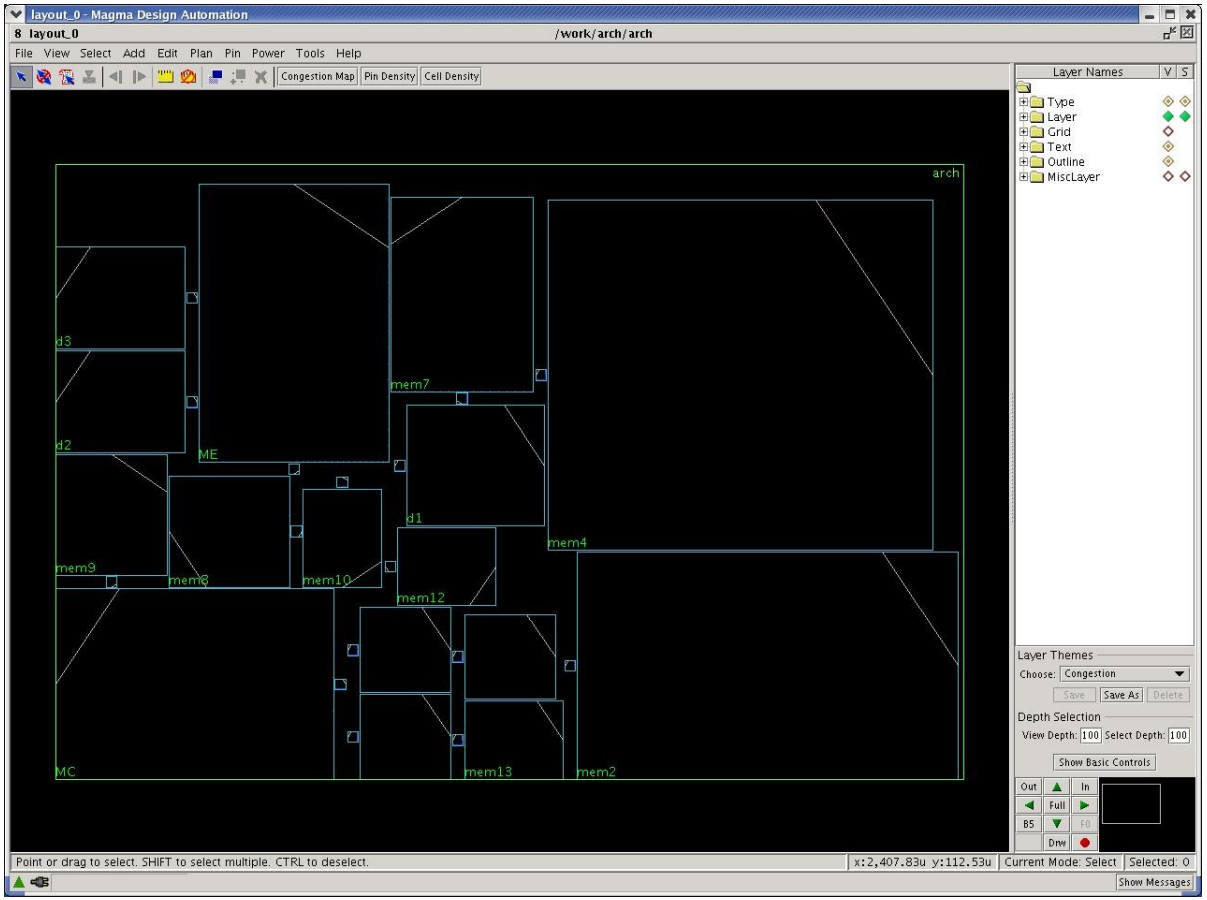


Floorplanner does not handle well blocks which have large difference in size



When the size of the switches is much smaller than the size of the memories, Parquet (academic macro block floorplanner) can not place the switches in appropriate locations.

Solution: add the switches after floorplanning



Step1: Parquet generates floorplan without switches

Step2: switches are added and imported to MAGMA:

- Close to the communication ports
- Without overlapping to all the other blocks