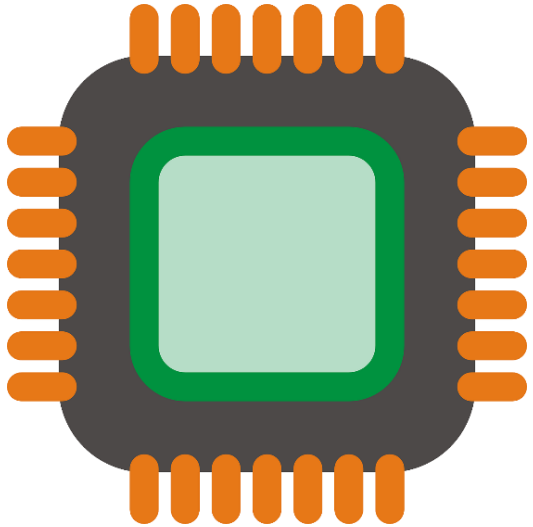
# LOW POWER SCHEDULING FOR HIGH LEVEL SYNTHESIS

Hardware Software Co-Design

# MOTIVATION



# WHY HIGH LEVEL SYNTHESIS (HLS)?

## Power consumption can be reduced by:

| | | | |
|---|---|---|---|
| Reducing chip and package capacitance → Expensive. | Scaling the supply voltage → Need extra circuits. | Better design techniques → Cost-effective. | Power management strategies → significant power savings. |

## Solution: Integrate power scheduler in High Level Synthesis (HLS) [1].

# WHAT IS HLS?

- High Level Synthesis: Translation function from behavioural to structural description [3].
- HLS consists of three phases [1]:
  - Scheduling: which clock cycle?
  - Allocation: how many resources?
  - Binding: which operation to which resource?

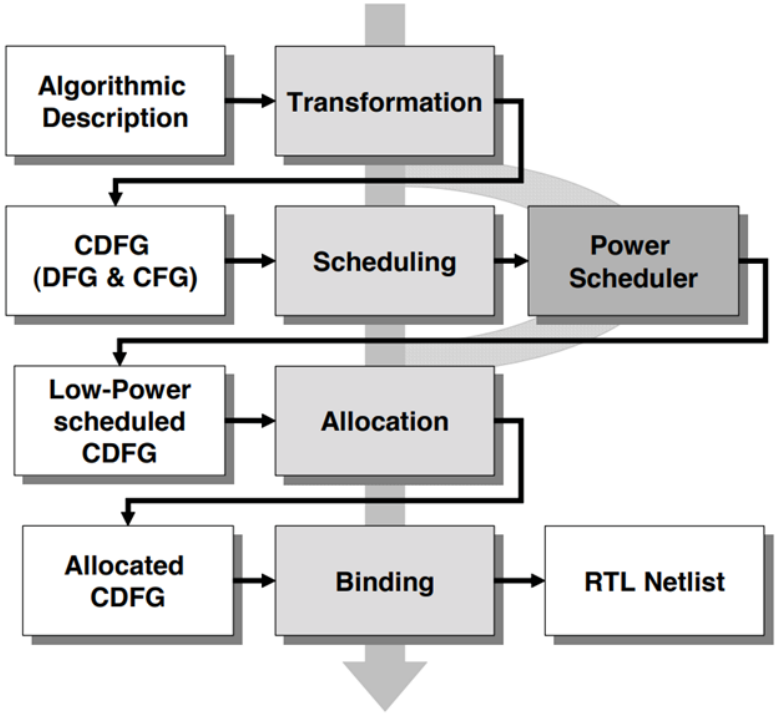



Fig. 1: HLS Design Flow with Power Scheduler integrated [1]

# SOURCE OF POWER DISSIPATION

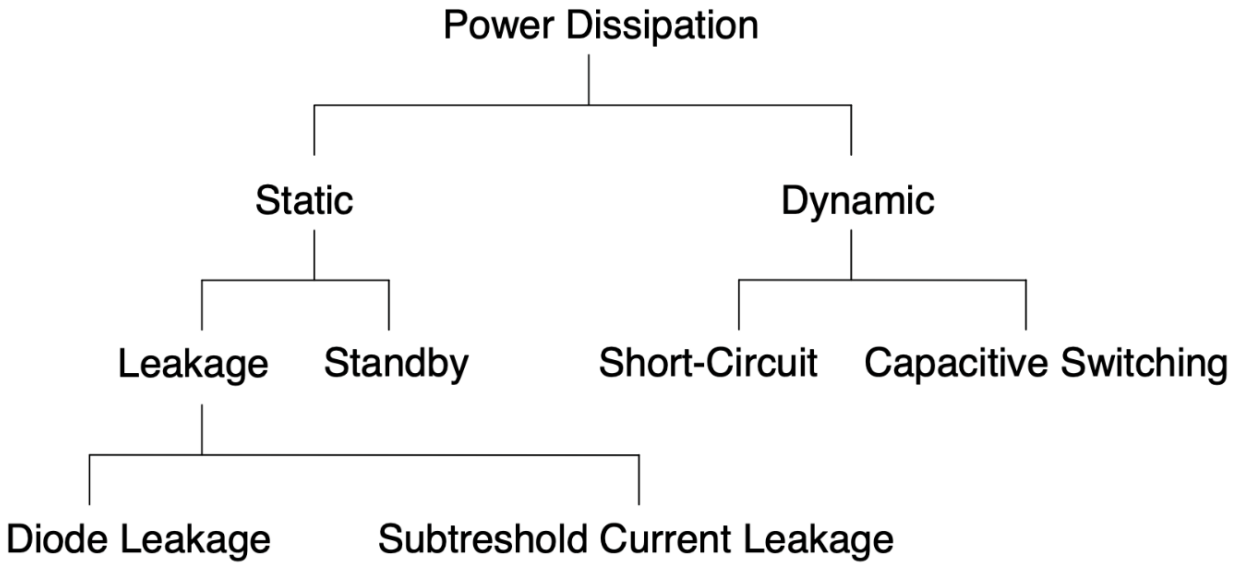



Fig. 2: Sources of power dissipation [1]

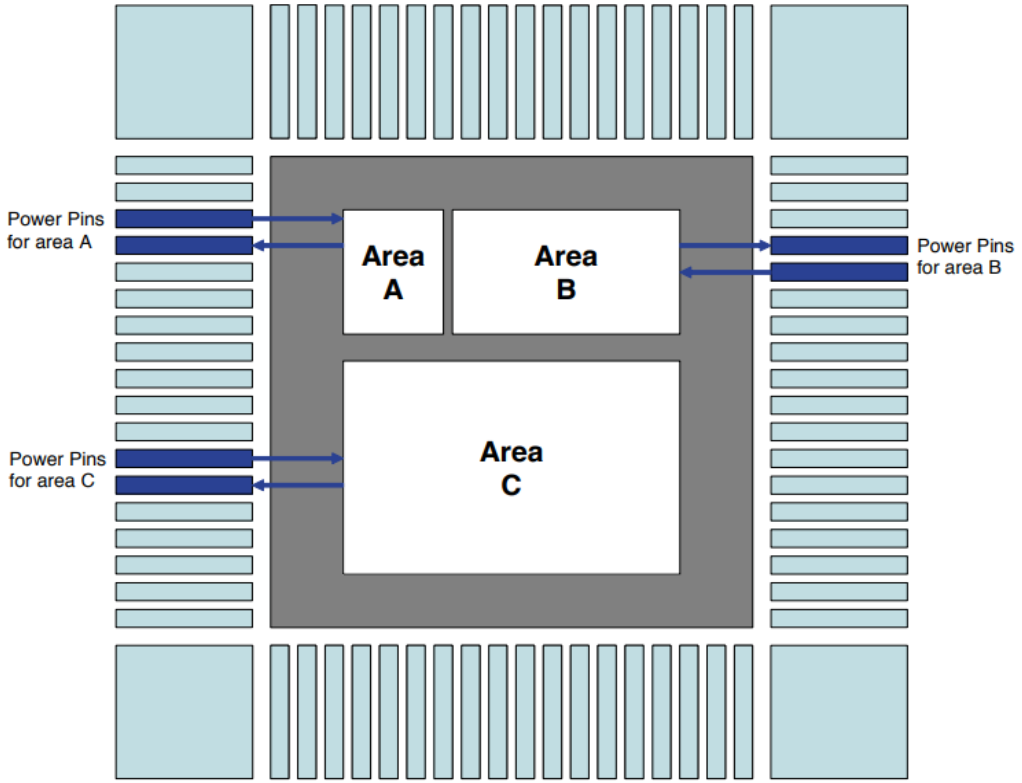



Fig. 3: Power down [1]

# STRATEGY

- **Goal 1**: Dedicated turn-on and turn-off mechanisms.
- **Goal 2**: Minimizing control components for those mechanisms.
- **Technique**: Power down.
- **Scheduling method**: Partitioning.

# FLOW OF POWER SCHEDULER

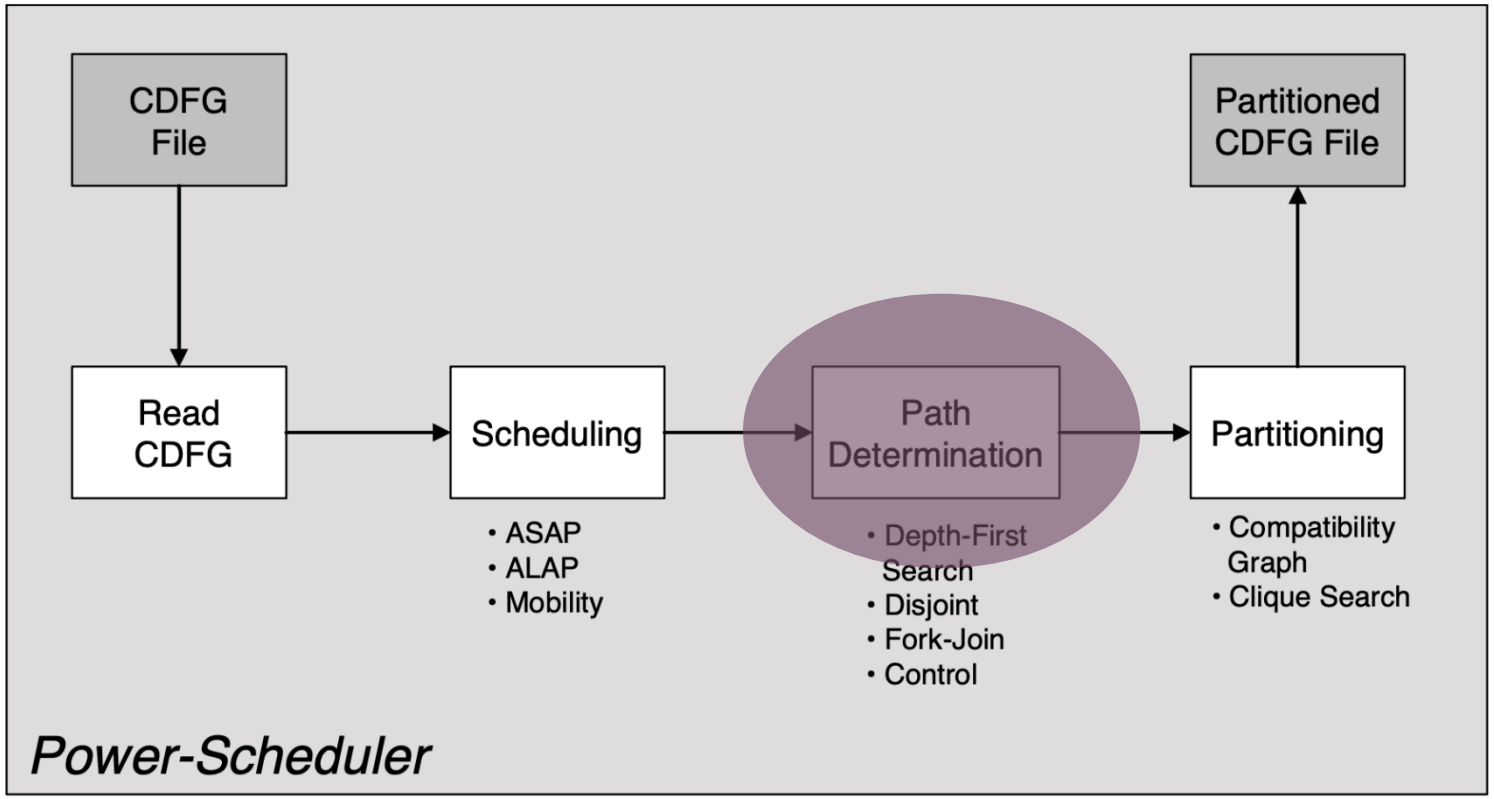



Fig. 4: Steps of Power Scheduler [1].

# DISJOIN PATH CALCULATION

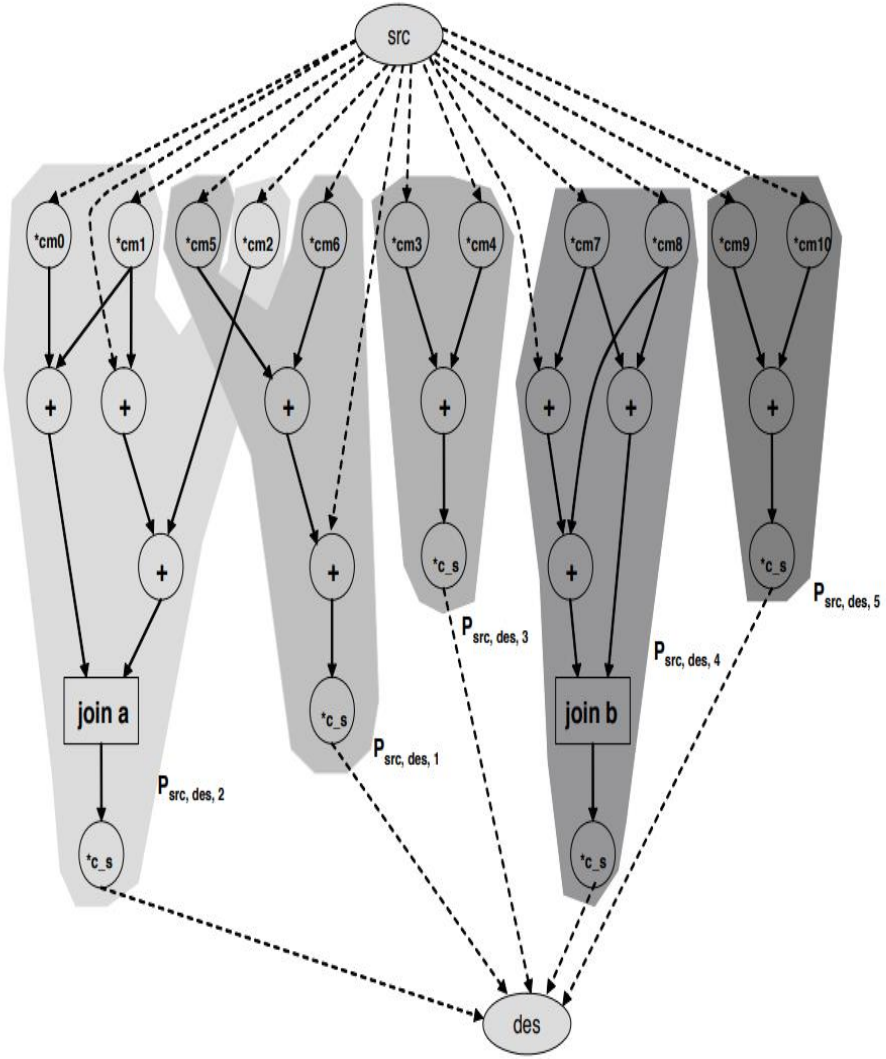


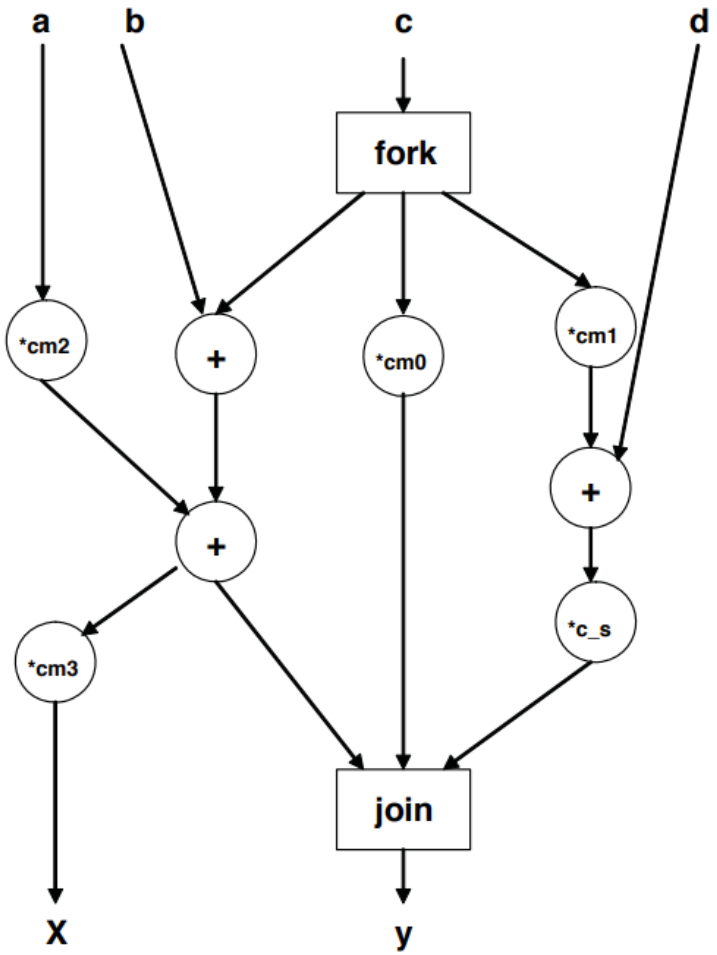

Fig. 5: Fork-join path analysis [1].

- * : 2 timesteps
- + : 1 timestep.

| Name | Path | Time |
|---|---|---|
| $P_{src,des,1}$ | {*cm5,*cm6, + , +,*c_s} | 4 |
| $P_{src,des,2}$ | {*cm0,*cm1 , *cm2, + +, +, join a, *c_s} | 5 |
| $P_{src,des,3}$ | {*cm3,*cm4 , +,*c_s} | 3 |
| $P_{src,des,4}$ | {*cm7, *cm8, + , +, +, join b, *c_s} | 5 |
| $P_{src,des,5}$ | {*cm9,*cm10 , +,*c_s} | 3 |

Table 1: Determined path from disjoint paths [1].

# FORK-JOIN PATH CALCULATION




- * : 2 timesteps
- + : 1 timestep.

| Name | Path | Time |
|---|---|---|
| $P_{fork,join,1}$ | {*cm2, + , +} | 4 |
| $P_{fork,join,2}$ | {*cmo} | 2 |
| $P_{fork,join,3}$ | {*cm1,+,*cs} | 5 |

Table 2: Determined path from fork-join paths [1].

Fig. 6: Fork-join path analysis [2].

# CONTROL PATH CALCULATION

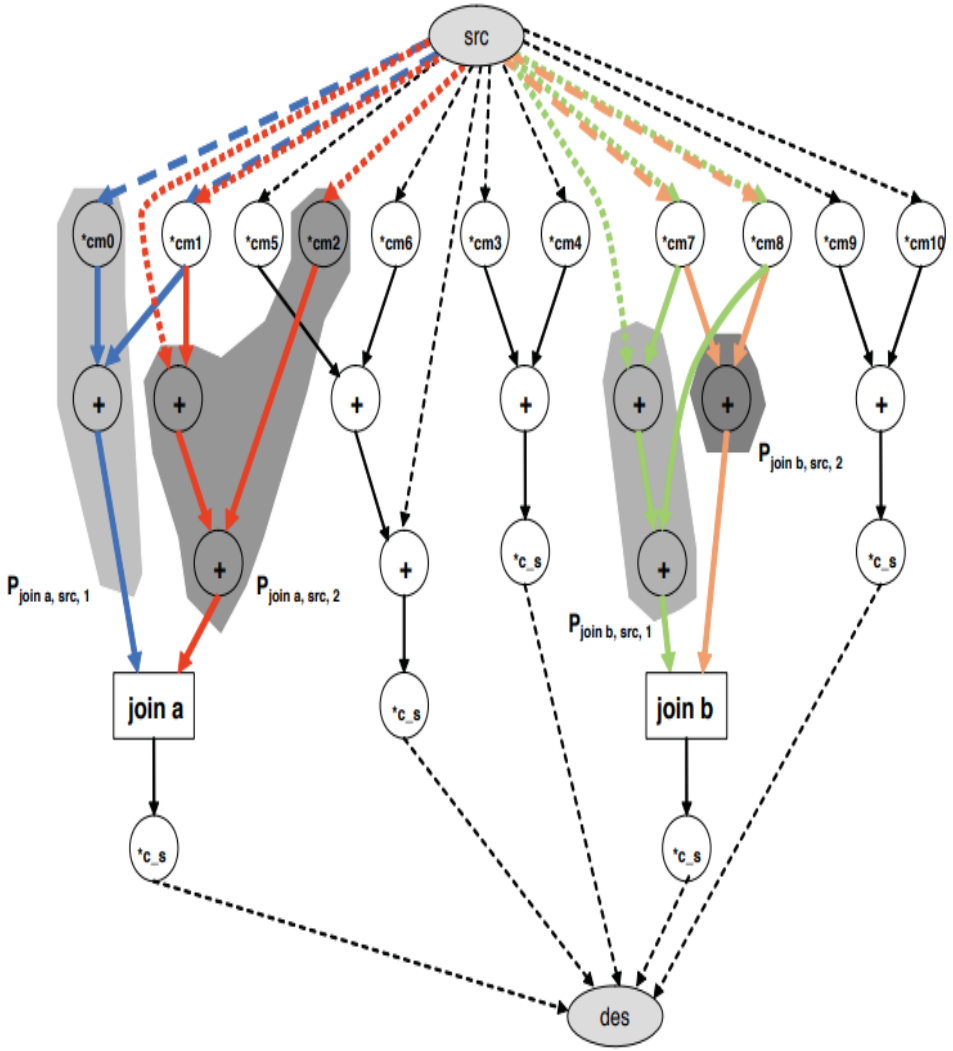



Fig. 7: Control path analysis [1].

- * : 2 timesteps
- + : 1 timestep.

| Name | Path | Time |
|---|---|---|
| $P_{join\ a,src,,1}$ | {*cm0, +} | 2 |
| $P_{join\ a,src,,1}$ | {*cm2, +, +} | 2 |
| $P_{join\ b,src,,1}$ | +, +} | 2 |
| $P_{join\ b,src,,1}$ | {+} | 1 |

Table 3: Determined path from control paths [1].

# COMPATIBILITY GRAPH

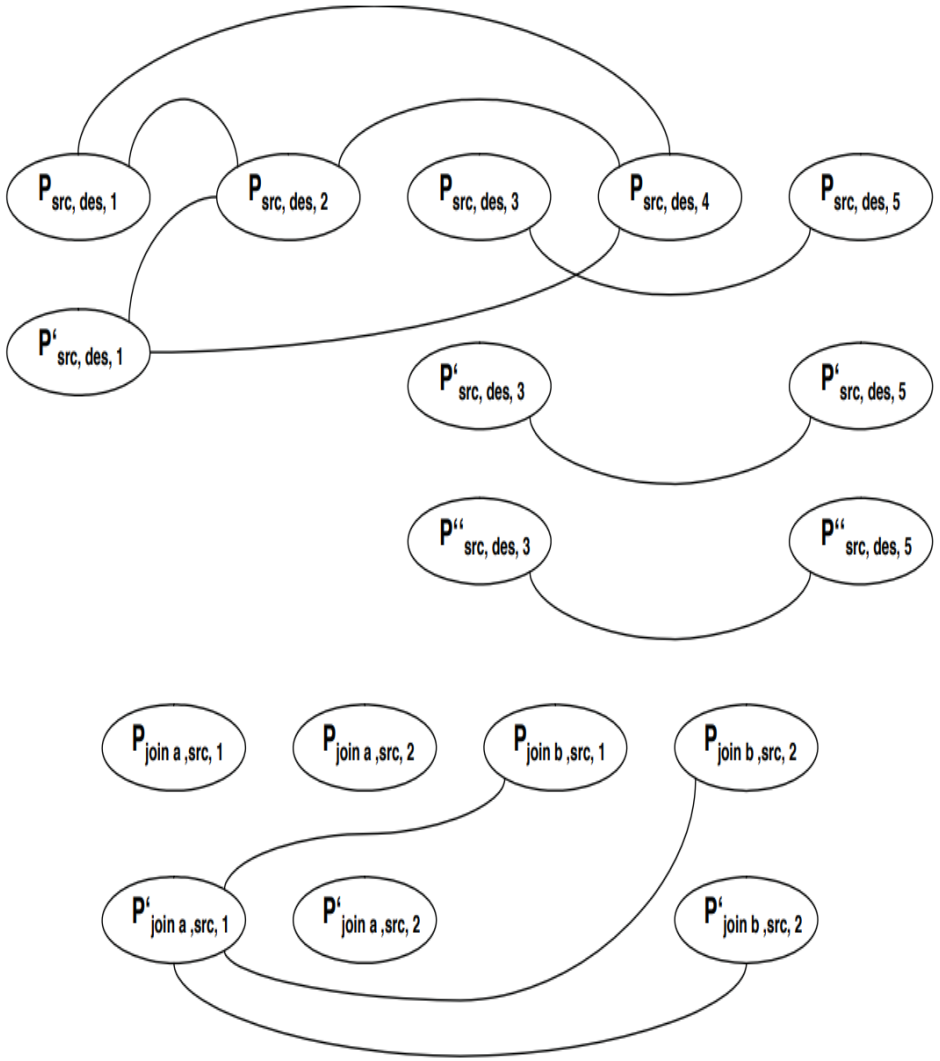
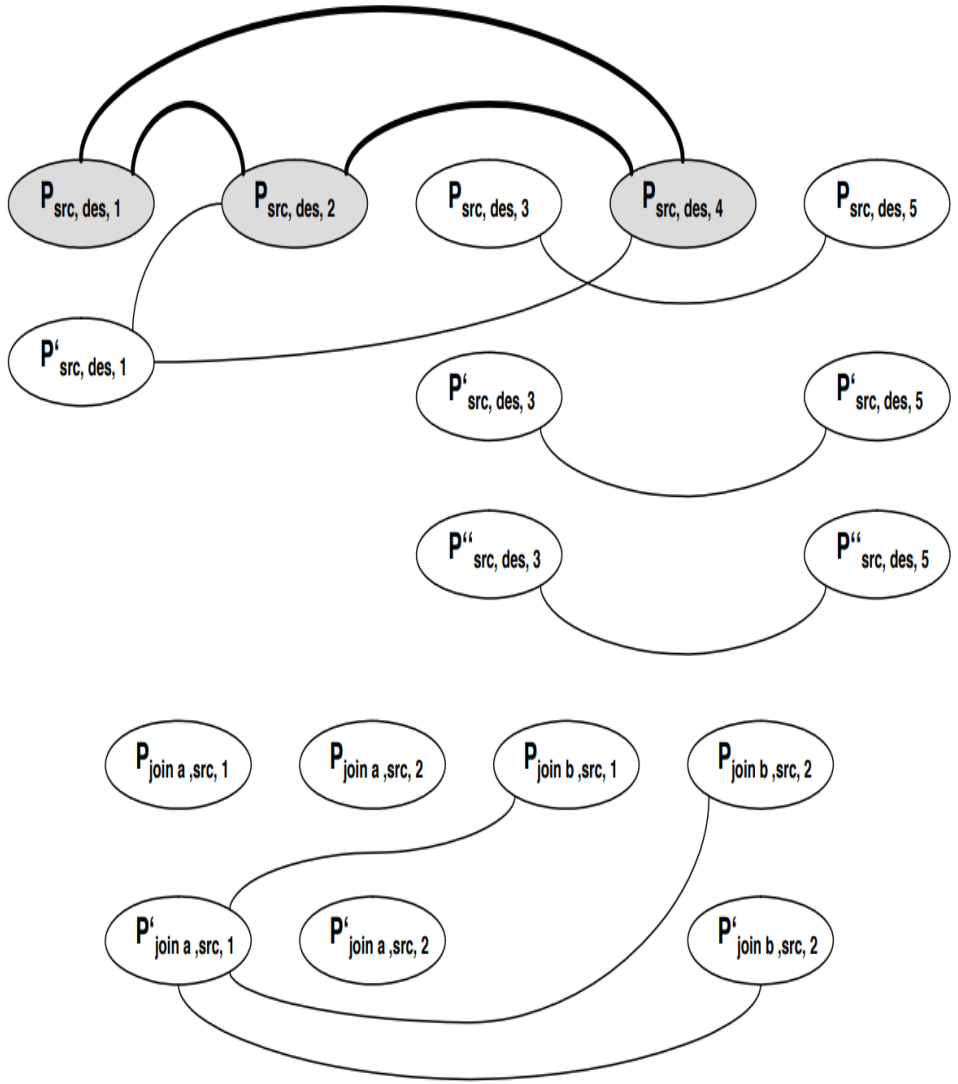



Fig. 8: Formation of compatibility graph [1].

# CLIQUE SEARCH




| Name | Partition |
|---|---|
| A1 | $P_{src,des,1}$ , $P_{src,des,2}$ , $P_{src,des,4}$ |
| A2 | $P_{src,des,3}$ , $P_{src,des,5}$ |
| A3 | $P_{join\ b,\ src,1}$ , $P'_{join\ a,src,1}$ |
| A4 | $P_{join\ a,\ src,2}$ |
| A5 | $P_{join\ b,\ src,2}$ |

Table 4: Partitions determined from clique search [1].

Fig. 11: Clique search from compatibility graph[1].

# OVERVIEW AND CONCLUSION

| | Partitioned | Unpartitioned |
|---|---|---|
| Power consumption | 1008000 µJ | 1180000 µJ |

- Low power driven synthesis is discussed.
- A developed power scheduler is the focus.
- Scheduler produces partitioned CDFG.
- Partition allows for dedicated turn-on and turn-off mechanism.
- Part of MPEG-2 algorithm is used as an example.
- Energy saving of 15% achieved [4].

# REFERENCES


[1] Achim Rettberg, Bernd Kleinjohann, and Franz Rammig. “Low Power Driven High-Level Synthesis for Dedicated Architectures”. In: (Jan. 2006).

[2] Achim Rettberg and Franz Rammig. “Integration of Energy Reduction into High-Level Synthesis by Partitioning”. In: From Model-Driven Design to Resource Management for Distributed Embedded Systems. Ed. by Bernd Kleinjohann et al. Boston, MA: Springer US, 2006, pp. 225–234. ISBN: 978-0-387-39362-9.

[3] M.C. McFarland, A.C. Parker, and R. Camposano. “The high-level synthesis of digital systems”. In: Proceedings of the IEEE 78.2 (1990), pp. 301–318. DOI: 10.1109/5. 52214.

[4] A. Rettberg and F.J. Rammig. “A new design partitioning approach for low power high-level synthesis”. In: Third IEEE International Workshop on Electronic Design, Test and Applications (DELTA’06). 2006, 6 pp.–148. DOI: 10. 1109/DELTA.2006.8.