LOW POWER SCHEDULING FOR HIGH LEVEL SYNTHESIS

Hardware Software Co-Design

MOTIVATION

- Key drivers of digital chip design: area, performance and power.
- Increasing demand for personal computing devices.
- Hence increasing demand for low-power.
- Challenges in traditional approach design:
    - Time consuming
    - Unreliable.
- Power issues must be addressed early in design cycle (system level) [1].

MOTIVATION

- 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].

BACKGROUND

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

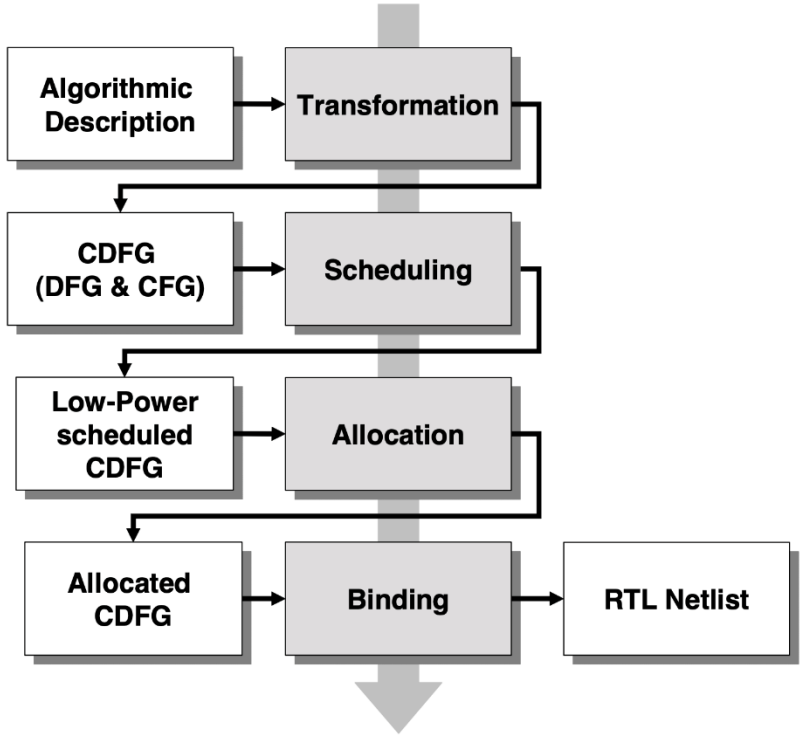

Fig. 1: HLS Design Flow [1]

BACKGROUND

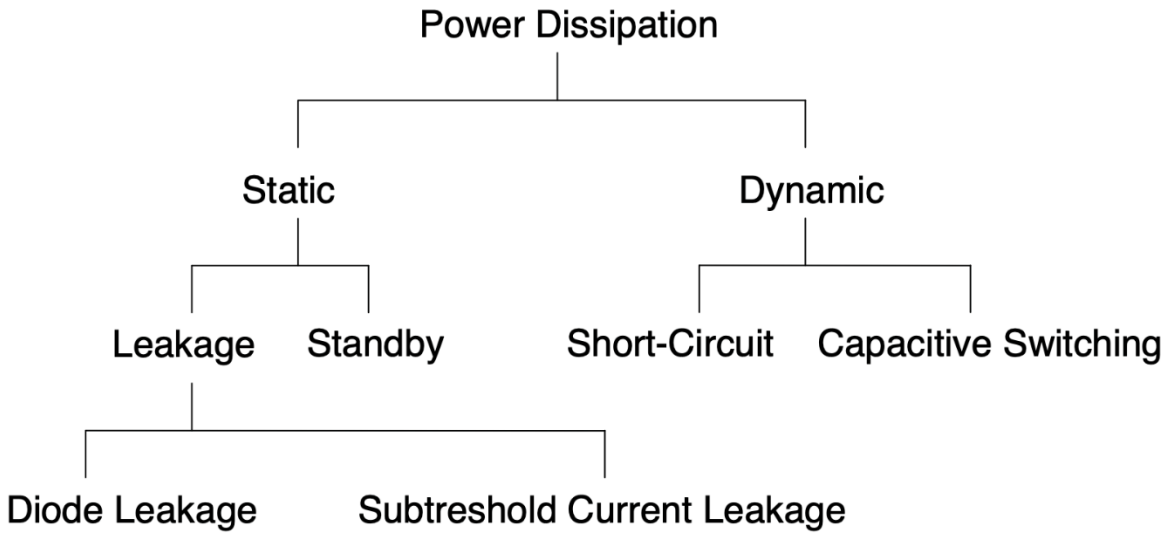

Fig. 2: Sources of power dissipation

- Short-circuit power dissipation accounts for less than 20% of total dynamic power [1].
- Focus: Capacitive switching.

$$P_{dynamic} = \frac{1}{2}\alpha C_L {V_d}^2 f$$

BACKGROUND

- Power Reduction Methods

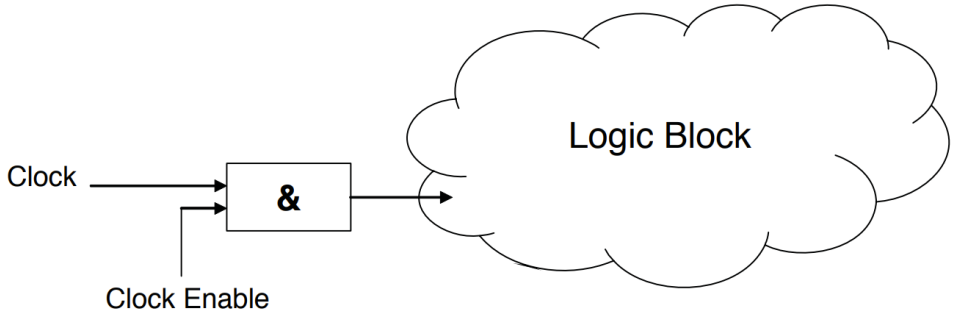

Fig. 3: Gate clock [1]

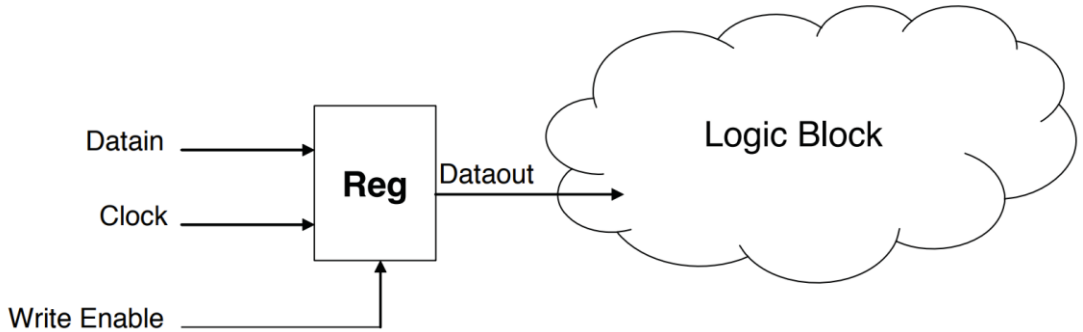

Fig. 4: Guarded evaluation [1]

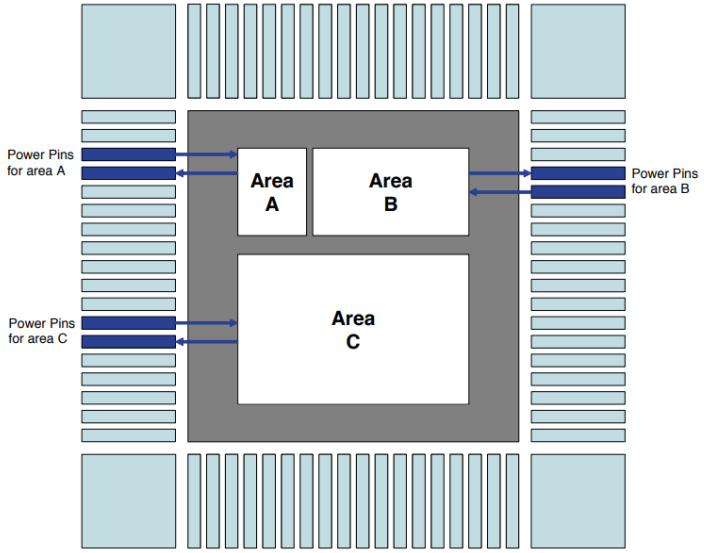

Fig. 5: Power down [1]

STRATEGY

- Goal 1: Dedicated turn-on and turn-off mechanisms.
- Goal 2: Minimizing control components for those mechanisms.
- Technique: Clock gating, power down or write enables for guard.
- Exploitation of idleness of device.
- Integration of low power method in scheduling process.

BACKGROUND

- Power Scheduler: alternative scheduling technique for power reduction.
- Path-based scheduling technique.

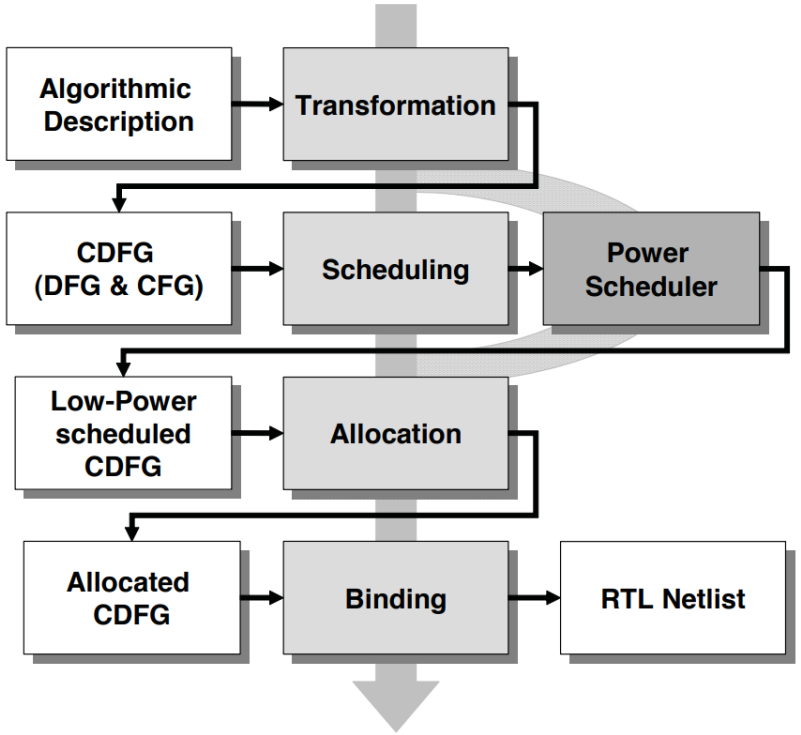

Fig. 6: Integration of Power scheduler in HLS design flow. [1]

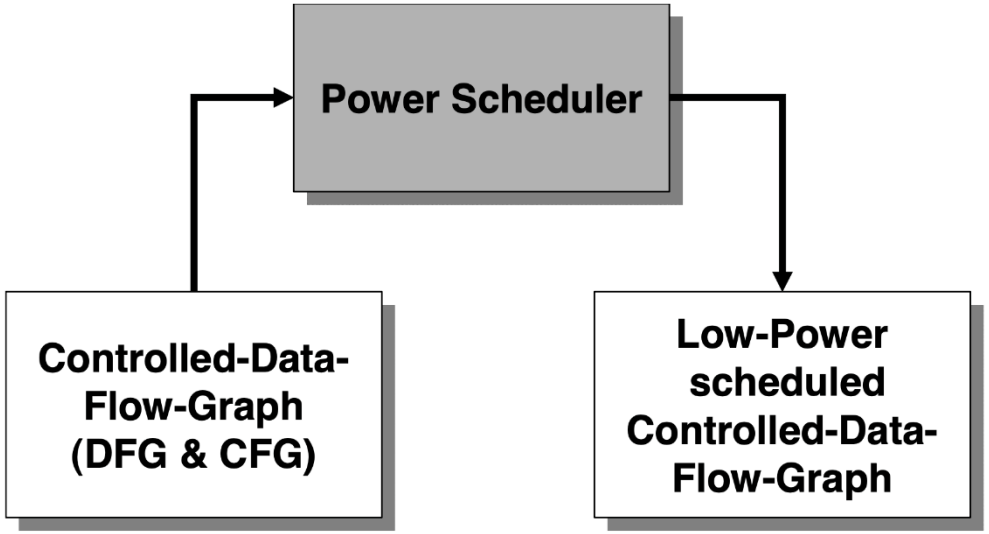

Fig. 7: Input and output of Power Scheduler [1].

BACKGROUND

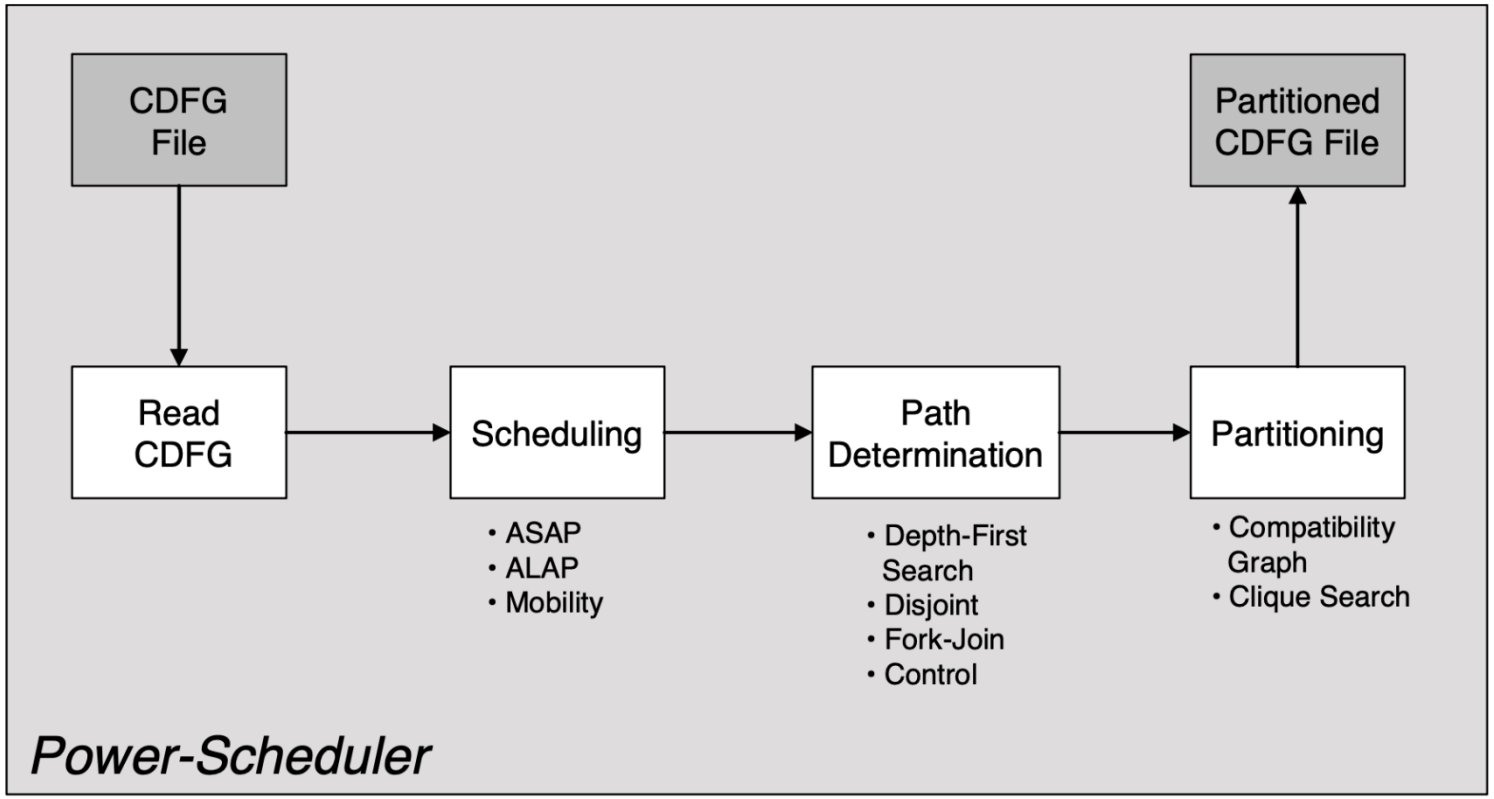

Fig. 8: Steps of Power Scheduler [1].

MATHEMATICAL IMPLEMENTATION

- Total Dynamic power of all nodes: $\sum_{i=1}^{n} P_{dyn,i}$
- Dynamic power for a partition: $\sum_{\mathrm{j}=1}^{n} P_{dyn,j}$
- Total dynamic power dissipation (with control unit): $\sum_{\mathrm{j}=1}^{p} P_{dyn,p} + P_{\mathrm{gc},p}$
- Power Delay of a partition (with run-time): $PD_{\mathrm{k}} = \left(P_{\mathrm{dyn},k} * d_{\mathrm{k}}\right) + \left(P_{\mathrm{gc},k} * d_{\mathrm{i}}\right)$
- Total power delay: $\sum_{x=1}^{l} PD_x$

FORK-JOIN PATH CALCULATION

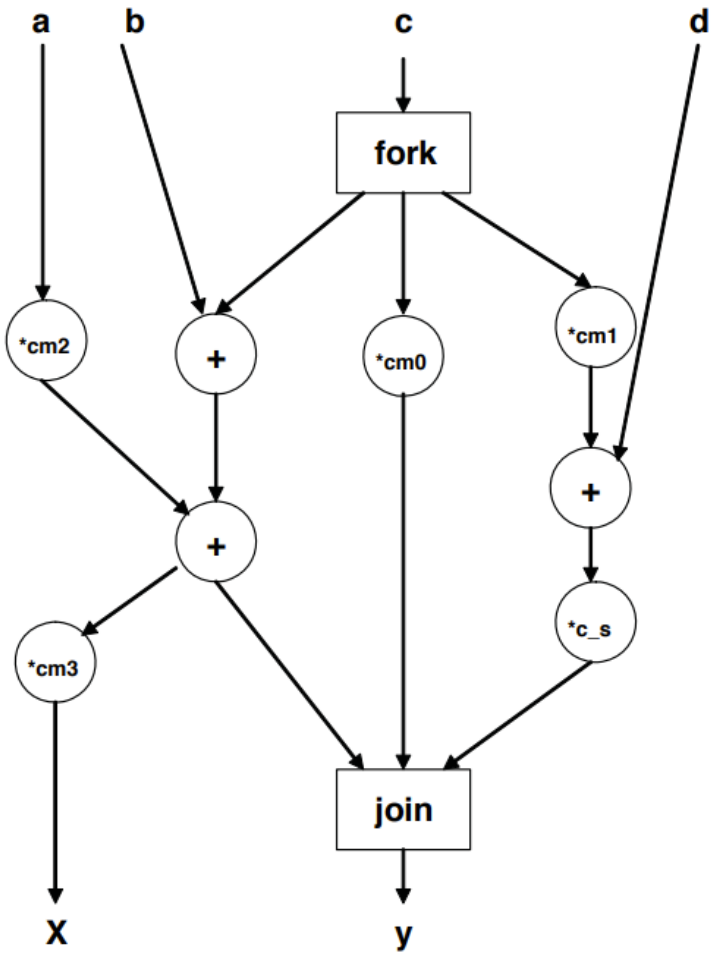

Fig. 9: Fork-join path analysis [2].

- * : two timesteps, + : 1 timestep.
- $P_{fork,join,1}$ = {*cm2, + , +}, time($P_{fork,join,1}$) = 4
- $P_{fork,join,2}$ = {*cmo}, time($P_{fork,join,2}$) = 2
- $P_{fork,join,3}$ = {*cm1,+,*cs}, time($P_{fork,join,3}$) = 5

CONTROL PATH CALCULATION

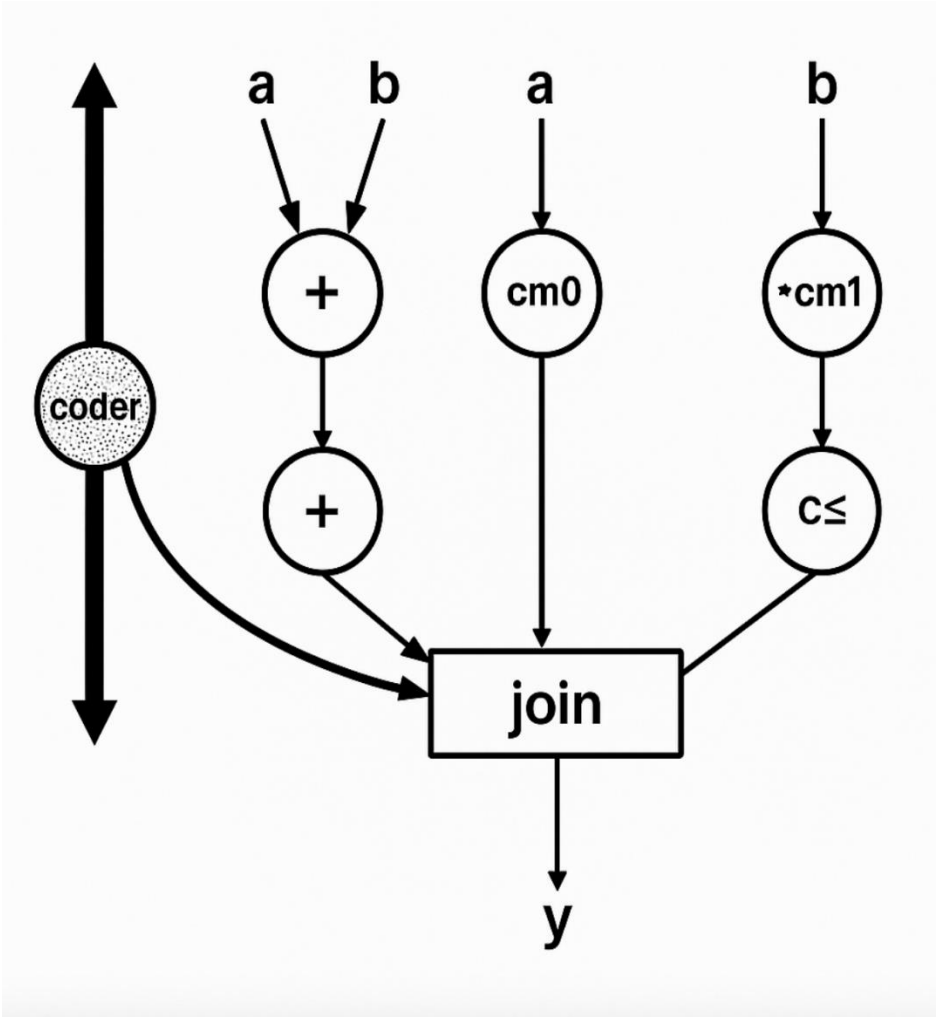

Fig. 10: Control path analysis [2].

- * : two timesteps, + : 1 timestep.
- $P_{+,join,1}$= {+ , +}, time($P_{+,join,1}$) = 2
- $P_{*cm0,join,1}$ = {*cmo}, time($P_{*cm0,join,1}$) = 2
- $P_{*cm1,join,1}$= {*cm1,+,*cs}, time($P_{*cm1,join,1}$) = 4

IMPLEMENTATION

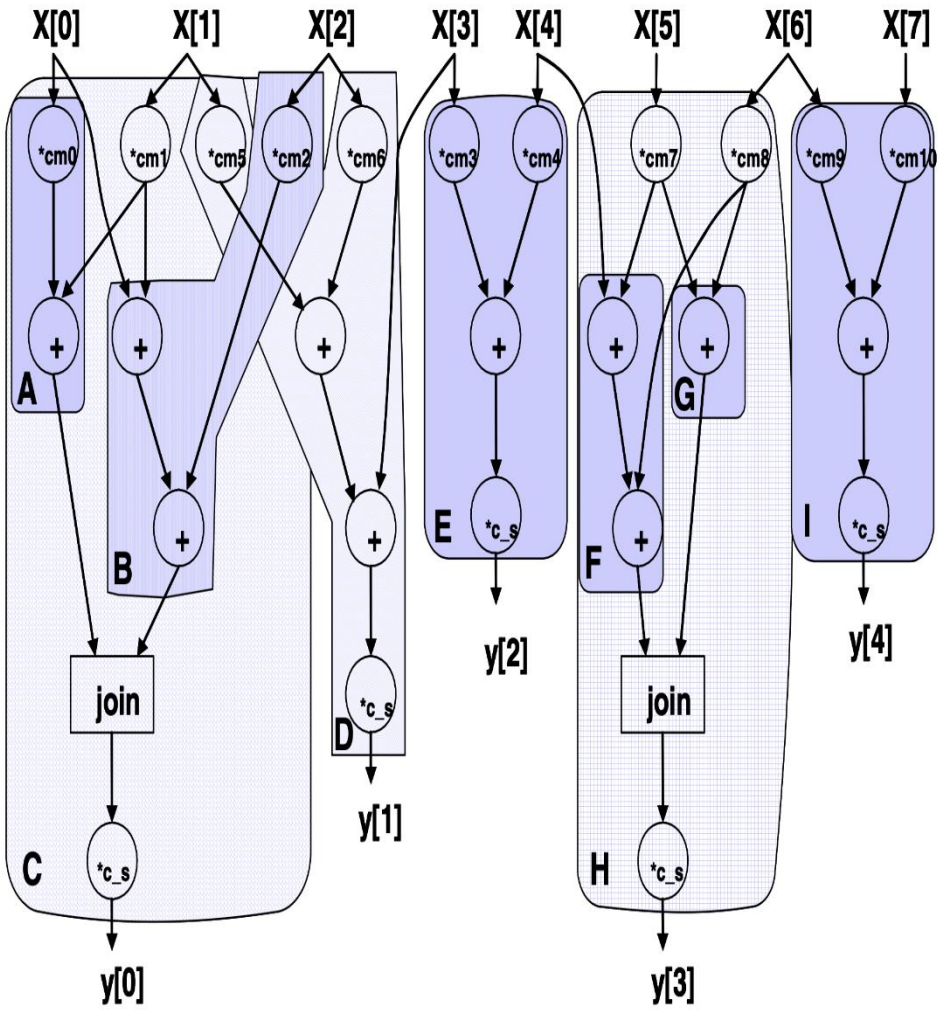

Fig. 11: Mpeg-2 algorithm example to determine partitions [1].

- Partitions:
- $P_1 = (A, F, G)$
- $P_2 = (C, D, H)$
- $P_3 = (E, I)$

OVERVIEW AND CONCLUSION

- 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 Parti- tioning”. 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.