Structural Timing-Aware Circuit Partitioning with FeasibilityConstraints for Multi-Chiplet Design

Hengyuan Zhang*, KangLin Tian*, Zirui Li, Jianwang Zhai†, XiuLi Fu and Kang Zhao
Beijing University of Posts and Telecommunications

Introduction

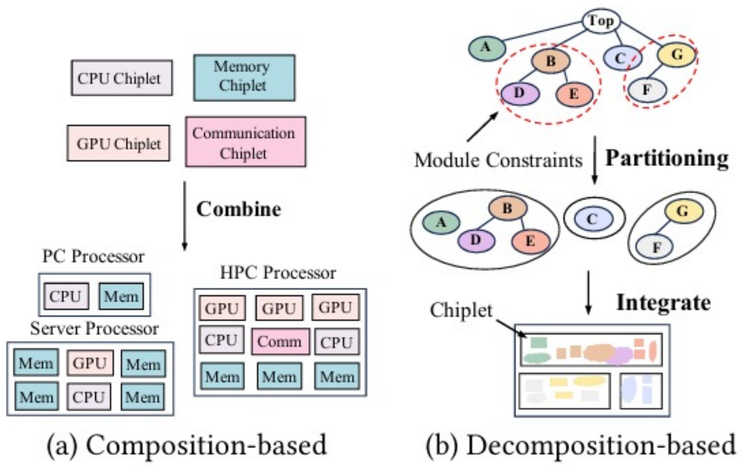

Figure 1. Two Chiplet Design Paradigms

Background & Challenge

- Chiplet partitioning determines intra-chiplet and inter-chiplet connections.
- Connectivity-only partitioning may cut timing-critical nets.
- Full timing analysis is too costly for early-stage partitioning.
- Hierarchy, module binding, and physical feasibility must be preserved.

Contributions

- **Constraint preservation:** hierarchy-aware module binding during hypergraph construction.
- **Timing guidance:** logic depth, load scale, and inter-module span encoded as hyperedge weights.
- **Physical validation:** post-placement TNS/WNS evaluation under realistic constraints.

Method

Framework

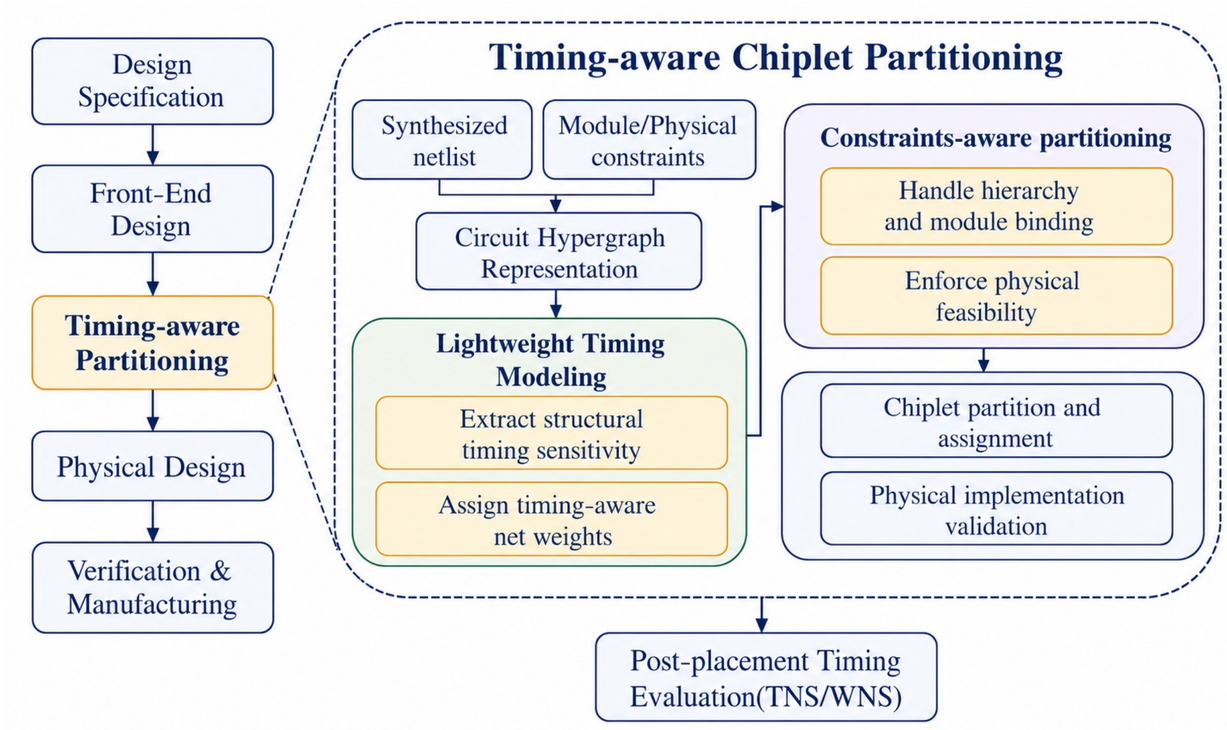

Figure 2. Overall Framework

The proposed framework follows four main steps:

- **Hypergraph construction:** convert the synthesized netlist into a circuit hypergraph.
- **Structural timing modeling:** extract net-level timing sensitivity from lightweight structural features.
- **Timing-aware weighting:** encode timing sensitivity into hyperedge weights for partitioning guidance.
- **Physical validation:** propagate chiplet assignments to placement and evaluate post-placement TNS/WNS.

Key Point

Constraint Handling

- Module binding constraints are interpreted from the design hierarchy.
- Constrained modules are contracted during hypergraph construction.
- Physical feasibility, including chiplet area and utilization, is checked during partitioning.

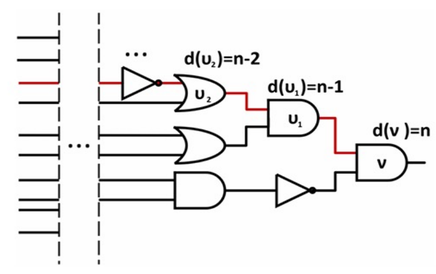

Figure 3. Logic Depth

Structural Timing Modeling & Weighting

- Timing sensitivity is estimated without invoking full static timing analysis.
- Three structural factors are used: **logic depth** (D_e), **load scale** (L_e), and **inter-module span** (S_e).

$$s(e) = \alpha \cdot D_e + \beta \cdot L_e + \gamma \cdot S_e, \quad \alpha + \beta + \gamma = 1$$

Higher s(e) → Larger Hyperedge Weight → Larger Cut Penalty

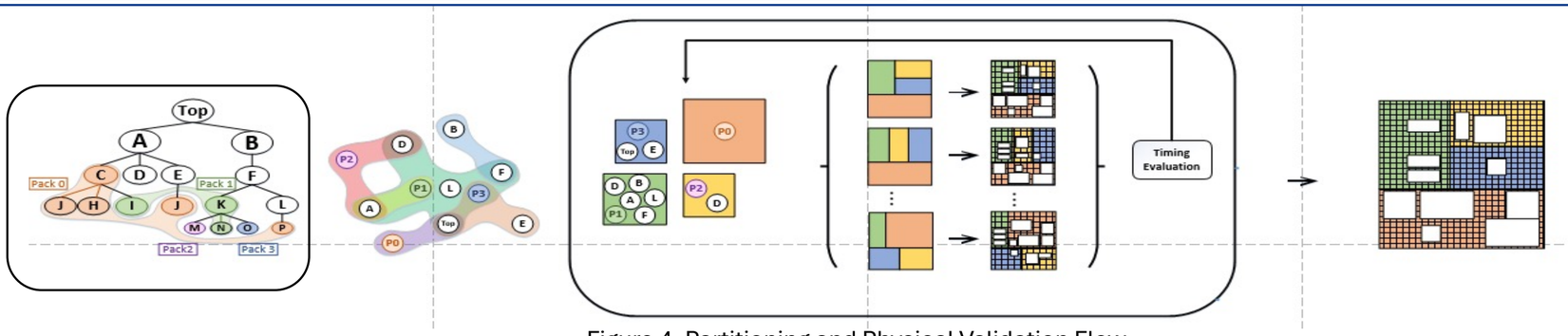

Figure 4. Partitioning and Physical Validation Flow

Result

- **Consistent timing improvement:** TNS is improved across KaHyPar, hMETIS, and PaToH.
- **Average TNS gains:** 13.2% on KaHyPar, 10.5% on hMETIS, and 22.1% on PaToH.
- **Better scalability with chiplet count:** timing benefit increases as more chiplet boundaries are introduced.
- **Key insight:** structural timing guidance helps avoid cutting timing-sensitive nets under realistic physical constraints.

Table 1. Post-placement Timing Comparison

Case	KaHyPar [6]		Proposed (Ours)		hMETIS [15]		Proposed (Ours)		PaToH [16]		Proposed (Ours)	
	TNS(ns)	WNS(ns)	TNS(ns)	WNS(ns)	TNS(ns)	WNS(ns)	TNS(ns)	WNS(ns)	TNS(ns)	WNS(ns)	TNS(ns)	WNS(ns)
case1	-83.847	**-2.980**	**-69.341**	-2.986	-145.862	-6.307	**-106.900**	**-4.985**	-107.211	-4.195	**-91.937**	**-2.927**
case2	-310251.380	-14.391	**-261220.510**	**-10.589**	-187346.630	-7.240	**-159209.555**	**-5.312**	-193499.223	-8.128	**-176979.627**	**-6.259**
case3	-1062.246	**-4.975**	**-944.488**	-7.561	-1419.855	**-3.554**	**-818.417**	-3.668	-1177.700	-3.083	**-903.668**	**-2.651**
case4	-210643.020	-11.410	**-199073.670**	**-9.250**	-161082.328	-6.139	**-156495.215**	**-5.932**	-198919.273	-6.960	**-144477.516**	**-5.208**
ratio	1.132	1.111	**1.000**	**1.000**	1.105	1.168	**1.000**	**1.000**	1.221	1.312	**1.000**	**1.000**

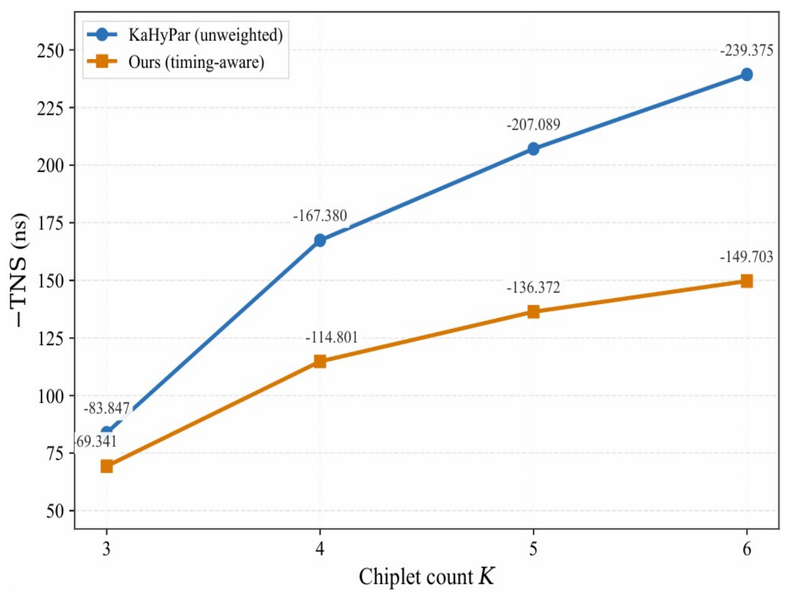

Figure 5. Impact of Chiplet Count