Intent Based Timing Constraints

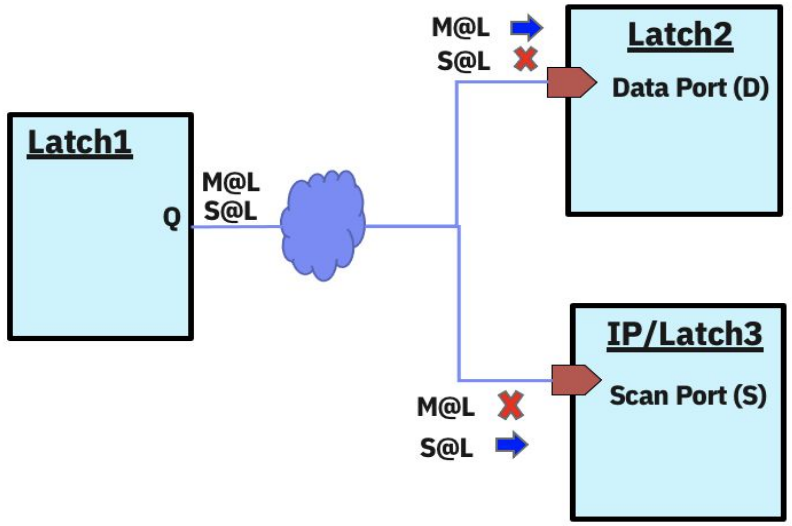
Hemlata Gupta
Adil Bhanji
Manish Verma
Jennifer Basile
Kerim Kalafala
Jack DiLullo

Motivation

- Achieve correct timing with minimum set of timing constraints
- Scenario: Only propagate functional clocks to functional pins
- Traditional approach: Apply don't care constraints for all non-functional clocks on functional pins

```
et::dont_care_signal –pin Latch2/D –phase S@L
et::dont_care_signal –pin Latch2/D –phase S1@L
...
et::dont_care_signal –pin Latch2/D –phase SMCw@T
```

```
et::dont_care_signal –pin Latch3/S –phase M@L
et::dont_care_signal –pin Latch3/S –phase Mx2@L
...
et::dont_care_signal –pin Latch3/S –phase AB_Mx8W@T
```



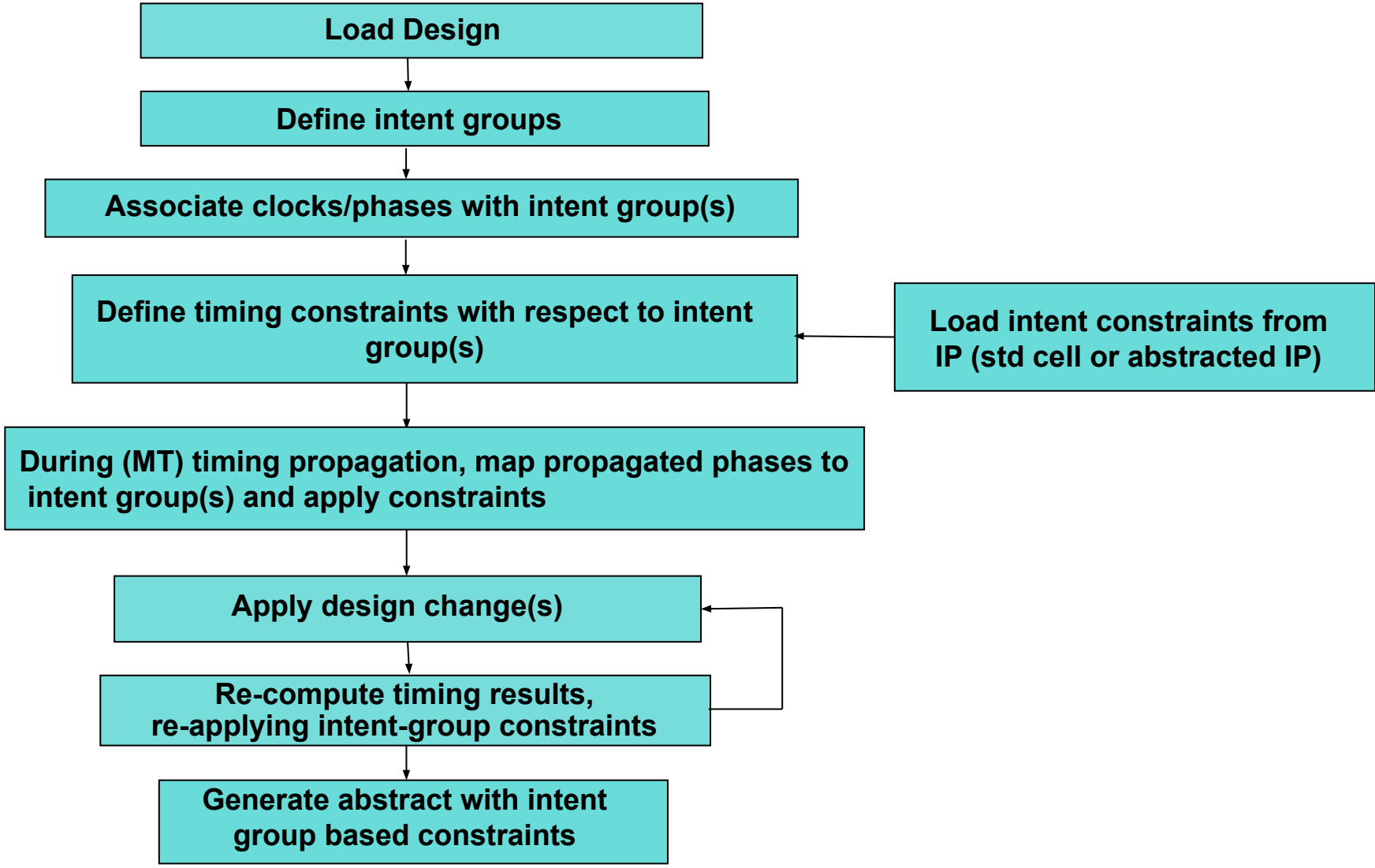
Challenges

- Increase in number of clocks in a design Increase in number of constraints
- Degradation of runtime and memory
- If applying constraints based on propagated signals, upfront signal propagation is required
- Additional clocks in flow Incorrect constraints, constraints are static/non-incremental
- Constraints get captured as is in abstract timing model large constraints file and large runtime/memory requirements for next level timing run

Solution Details

- Define clock intent groups
  - Example: functional, scan, test, etc.
- Assign design clocks to one or more intent groups
- Define timing constraints with respect to intent groups
- Tool action
  - propagated signals are mapped to clock intent groups
  - correct intent constraints are applied
- Capture intent groups and constraints based on intent-groups as part of abstract timing model
  - Use these constraints in upper level designs consuming abstract timing models

Timing Flow with “Intent Based” Timing Constraints



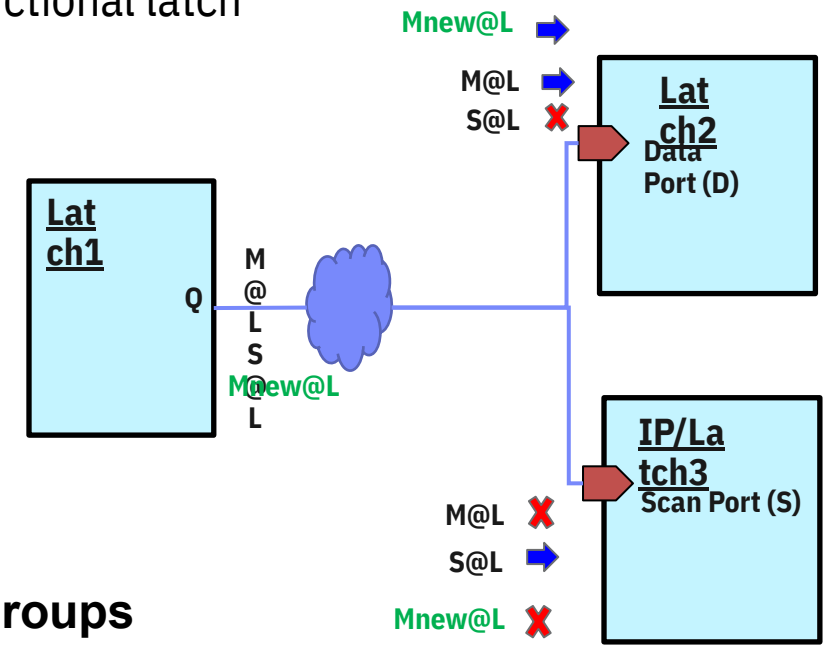
Example Scenario with “Intent Based” Timing Constraints

Scenario

- dont care scan phases on functional pin of a scan latch
- dont care functional phases on scan pin of a functional latch

et::dont_care_signal –pin Latch2/D –clock_types {scan@L scan@T}

et::dont_care_signal –pin Latch3/S –clock_types {functional@L functional@T}



”scan” and “functional” are new intent groups

Incremental constraints handling

New functional clock Mnew@L at Latch1/Q results in Mnew@L at Latch2/D and Latch3/S pins

- Dont care of Mnew@L at Latch3/S is done as part of incremental processing
- No extra constraint required for additional clocks

Benefits

Traditional method	Intent based method
Large number of constraints	**Small number of constraints**
Inefficient – Requires a throw-away timing propagation for applying constraints	**More efficient – Avoids the need for a throw-away timing propagation**
Non-incremental: Any change which alters the set of signals being propagated (constraints depend on these) causes a potential exposure	**Incremental: Any change which alters the set of signals being propagated handled naturally through incremental timing**

Potential for multi-hour savings on full chip timing

Small Macro Runs: Single Corner Freq Dependent Det Timing

Design	Checks Report Matching?	Audit Report Run Time		Audit Report Size		Total Run Time (Hr)	
		Current (sec)	New (sec)	Current	New	Current	New
MAC1	✓	573	41.1	320M	18M	1:13:46	0:52:20
MAC2	✓	123.9	13.3	61M	7.4M	0:20:50	0:18:05
MAC3	✓	20.5	9	16M	5.9M	0:08:59	0:12:52
MAC4	✓	2526.4	2848.5	1.5G	1.3G	2:05:36	2:19:33
MAC5	✓	895.6	26.9	213M	12M	0:58:36	0:40:30
MAC6	✓	114.2	17.5	49M	6.5M	0:54:40	0:32:02
MAC7	✓	50.6	18.2	34M	6.2M	2:40:27	2:58:04
MAC8	✓	209	14	121M	8.8M	0:37:26	0:26:39
MAC9	✓	573.8	46.6	157M	11M	0:45:35	0:50:55
MAC10	✓	29.8	8.9	18M	7.2M	0:10:39	0:08:46
MAC11	✓	12.9	6.8	5.8M	5.5M	0:12:22	0:07:58
MAC12	✓	291.2	46.8	145M	16M	0:41:58	0:31:31
MAC13	✓	153.8	64.7	66M	37M	0:20:12	0:13:28
MAC14	✓	110.5	12.7	64M	7.4M	0:21:45	0:17:08
MAC15	✓	334.1	27.6	202M	11M	0:39:39	0:38:22

Small Macro Runs: Freq Independent Statistical Timing

Design	Checks Report Matching?	Audit Report Run Time		Audit Report Size		Total Run Time (Hr)	
		Current (sec)	New (sec)	Current	New	Current	New
MAC1	✓	635.9	40.1	238M	18M	2:46:31	2:09:03
MAC2	✓	128.1	31.3	63M	7.8M	1:00:56	0:52:56
MAC3	✓	27.3	8.8	13M	5.9M	0:28:40	0:23:05
MAC4	✓	3984.1	3292.3	2.4G	2.2G	3:43:21	3:15:42
MAC5	✓	401.6	42.2	184M	15M	2:22:49	1:42:36
MAC6	✓	86.4	34.8	48M	17M	2:08:26	1:50:59
MAC7	✓	88.9	43.3	37M	17M	3:34:11	3:22:57
MAC8	✓	187.5	30.6	97M	11M	2:50:38	2:08:26
MAC9	✓	258.5	21.6	119M	11M	1:38:03	1:19:24
MAC10	✓	58.3	6.7	14M	5.7M	0:19:02	0:15:37
MAC11	✓	15.8	6.7	5.7M	5.4M	0:17:53	0:14:46
MAC12	✓	273.9	17.7	123M	9.8M	0:59:43	1:04:48
MAC13	✓	251.8	150.4	88M	60M	0:45:01	0:40:22
MAC14	✓	102.8	16.1	40M	8.4M	0:59:52	0:44:07
MAC15	✓	355.2	32.9	162M	14M	1:41:41	1:27:08

Unit Runs – Freq Dependent Deterministic Timing

Design	Setup Checks	Hold Checks	Runtime (Hr)		Memory (GB)		Audit Report Size (.gz)		Don't Care Count	
Unit1	✓	✓	01:55:03	01:43:20	67.79	64.85	3.6G	2.2G	130,955,134	15,265,519
Unit2	✓	✓	00:26:40	00:25:50	20.84	19.50	777M	175M	52,105,660	1,525,734
Unit3	✓	✓	00:26:24	00:26:05	22.66	22.44	512M	392M	35,367,616	25,143,242
Unit4	✓	✓	00:20:54	00:19:22	18.04	16.96	545M	100M	38.226,156	283,583
Unit5	✓	✓	00:35:11	00:33:48	28.80	28.52	726M	566M	53,541,785	40,635,044
Unit6	✓	✓	01:30:21	01:08:54	49.03	43.28	2.7G	394M	200,179,256	3,172,769

Audit report – Output file containing all constraints applied in a timing run