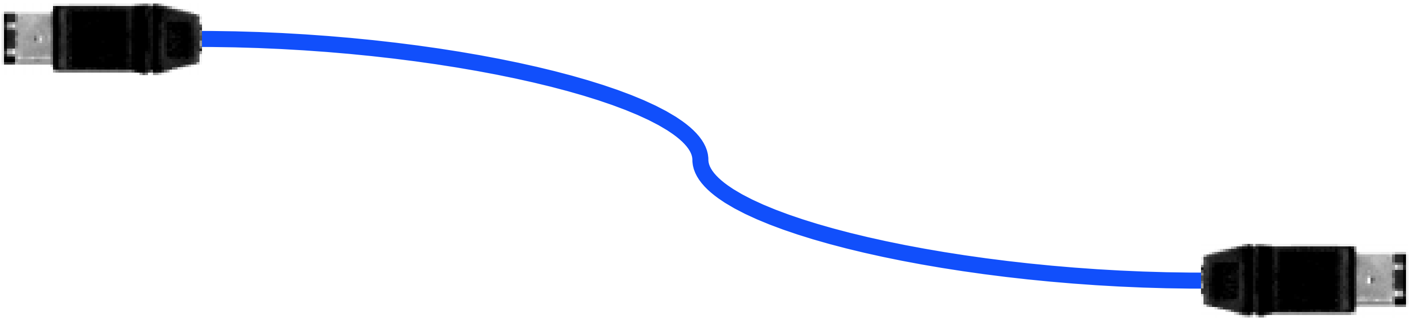
Synchronizing Cycle Master to External Timing Information via Cycle Slave



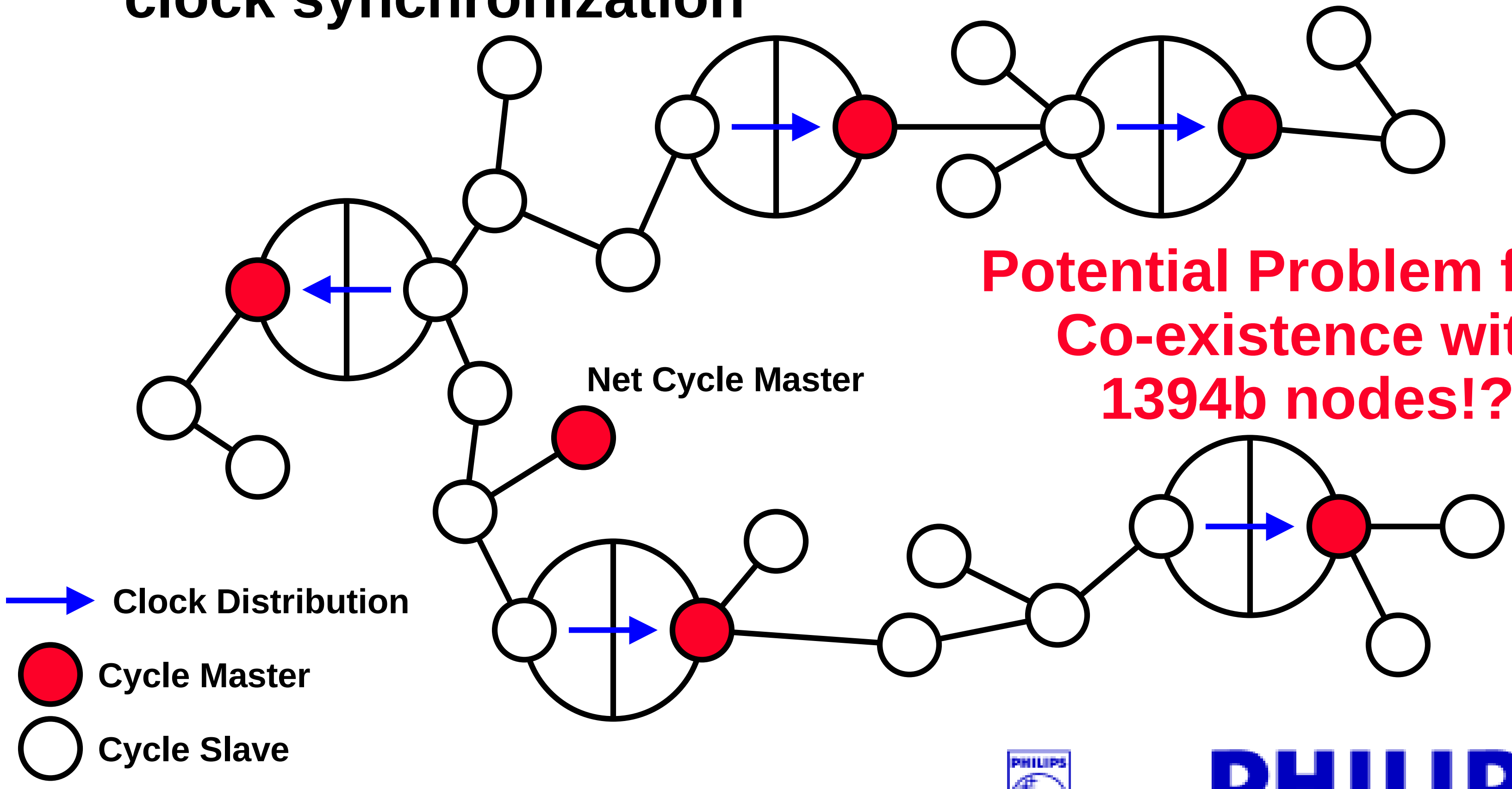
Solution for Cycle Master Location Limitation

Takashi Sato

Philips Research Briarcliff

Current Limitation

- **At least one of the portals of each Bridge has to be Cycle Master (root) to ensure network wide clock synchronization**



Potential Problem for Co-existence with 1394b nodes!?

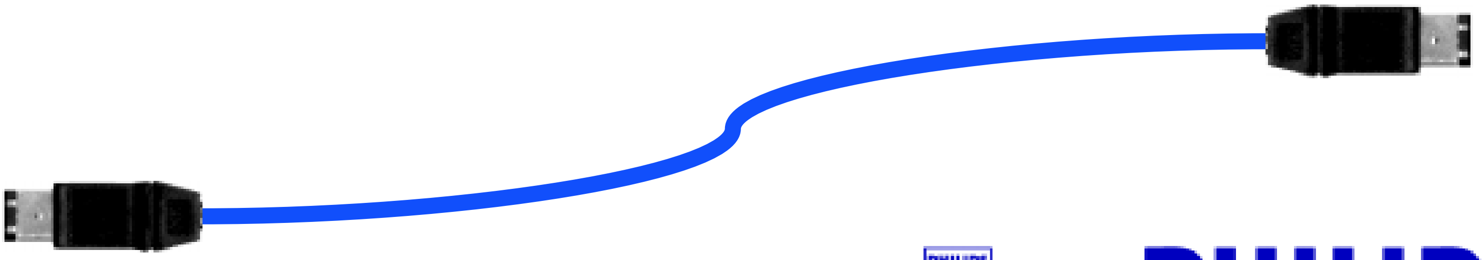
PHILIPS

Review - Cycle Time

- Cycle Time consists of

	Size	Range	Unit
– Second_Count:	7bits	[0, 127]	Second [s]
– Cycle_Count:	13bits	[0, 7999]	Cycle [125µs]
– Cycle_Offset:	12bits	[0, 3071]	Clock [40.7ns]

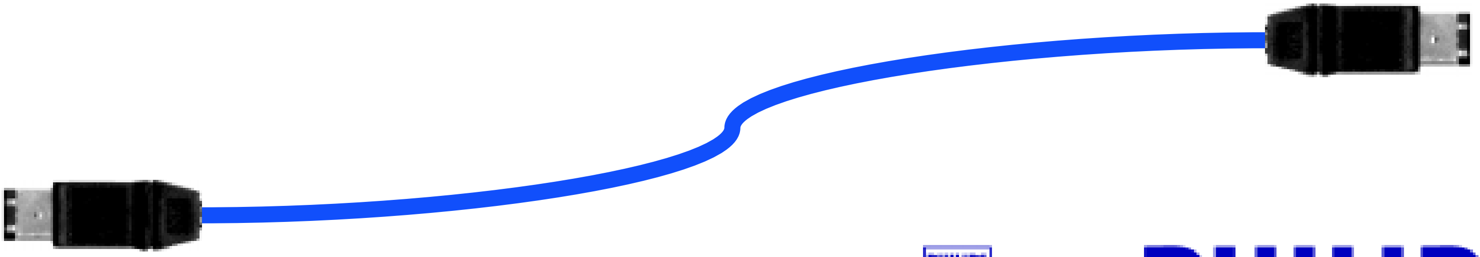
Synchronization of Cycle_Offset is the Most Difficult!



PHILIPS

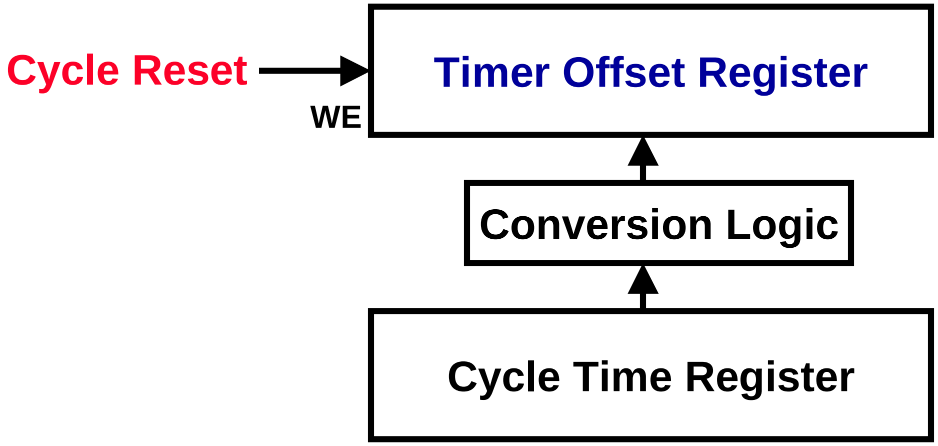
Solution

- **Implement Two Registers:**
  - **One for Cycle Slave: Timer Offset Register**
  - **One for Cycle Master: Timer Adjustment Register**



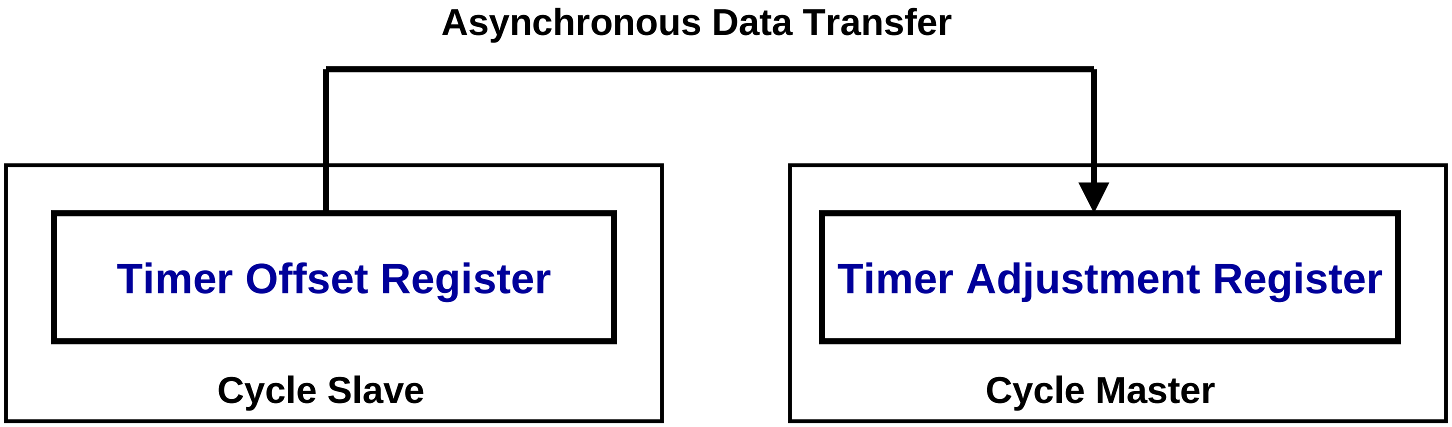
Timer Offset Register (in Cycle Slave node)

- Receive Cycle Reset (cycle_offset = 0) signal from outside local bus (the other portal)
- At Cycle Reset pulse, Capture the Cycle Offset field (12LSB) of Cycle Time Register as an integer value (e.g. 12 -> 12, 3070 -> -2) and store it into Timer Offset Register



Timer Offset Register -> Timer Adjustment Register

- Cycle Slave node reads Timer Offset Register value and sends it (write transaction) to Timer Adjustment Register in Cycle Master node
- This transfer of information can be performed automatically or controlled by microcontrollers

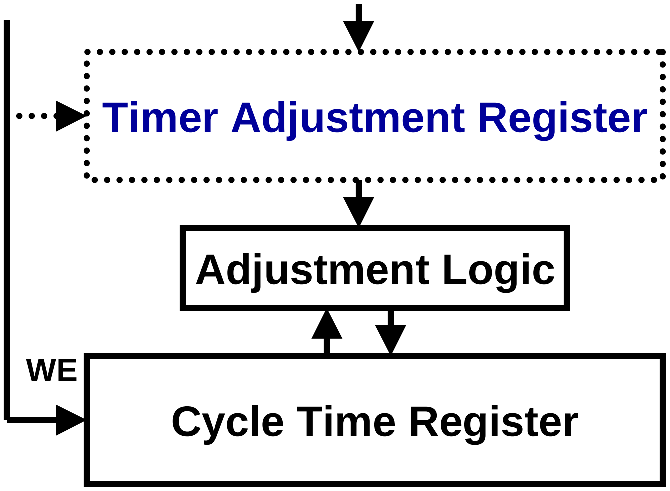


Timer Adjustment Register (in Cycle Master node)

- **Write operation to Timer Adjustment Register triggers the following operation**

  Cycle Time (new) = Cycle Time (old) - Timer Adjustment Value

- **Necessary adjustment (e.g. wrap-around process) is made to the resulted value**



Timer Adjustment Register

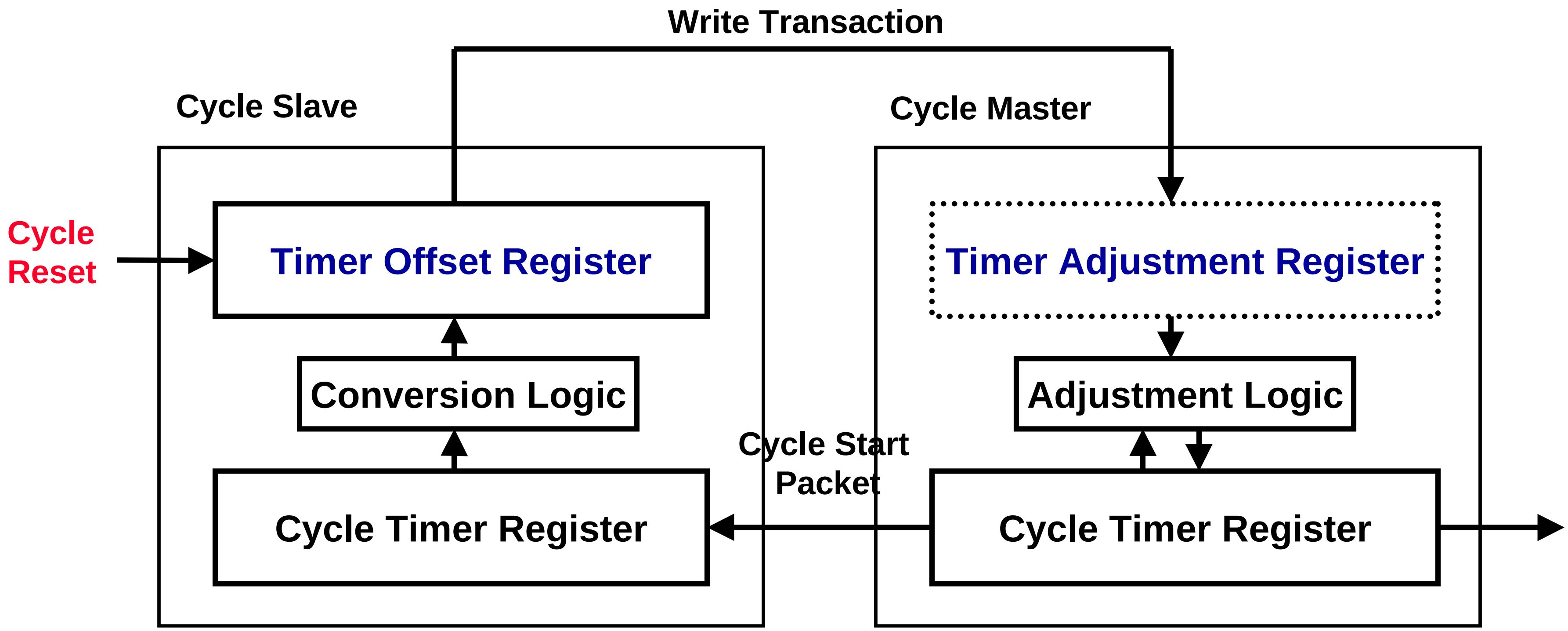
Adjustment Logic

WE

Cycle Time Register

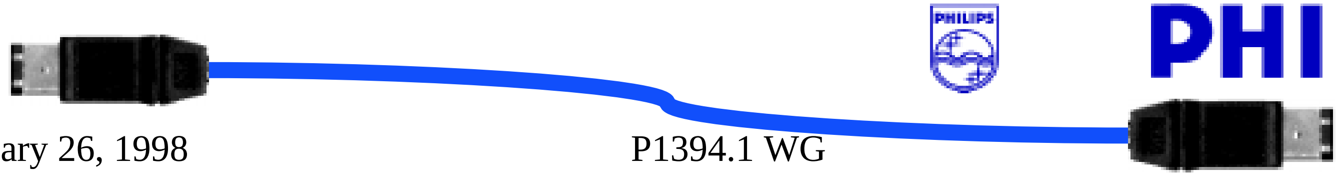
Cycle Synchronization Loop

Feedback Loop!



Advantages - Timer Offset/Adjustment Registers

- **Simple and Reliable: Robust against Packet Loss and/or Infrequent Update**
- **Efficient: No Need to Send any packet when Timer Offset Value is Zero**
- **Small Hardware Addition to New Silicon**
- **Backward Compatible: Software Solution available for Existing Silicon**



SW Solution - Timer Offset Register

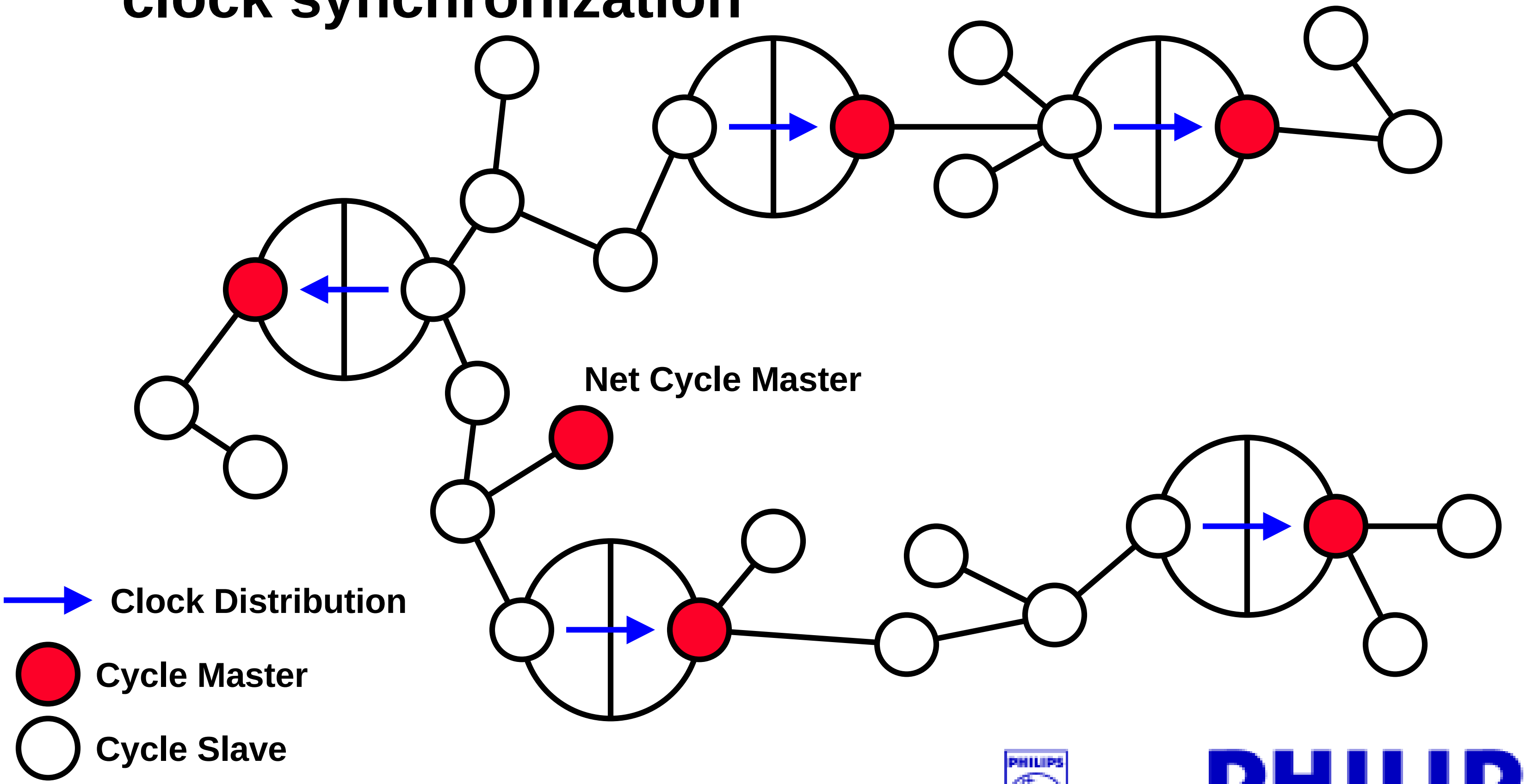
1. Detect Cycle Reset Interrupt
2. Read Cycle Offset field of Cycle Time Register
3. Subtract pre-determined value (processing delay between 1. and 2.) to get precise value of Cycle Time Register when Cycle Reset is asserted
4. If the value is greater than the threshold value (e.g., 1023), treat it as a negative number (subtract 3072)

SW Solution - Timer Adjustment Register

1. Intercept Write Transaction to Timer Adjustment Register Address
2. Read Cycle Time Register
3. Subtract Timer Offset Value that came in the write transaction
4. Add pre-determined value (processing delay between 2. and 6.) to get precise value of Cycle Timer when Cycle Reset is updated
5. Perform necessary adjustment process so that every field of Cycle Time fits in its proper range
6. Write the resulted value to Cycle Time Register

Current Limitation

- At least one of the portals of each Bridge has to be Cycle Master (root) to ensure network wide clock synchronization



Net Cycle Master

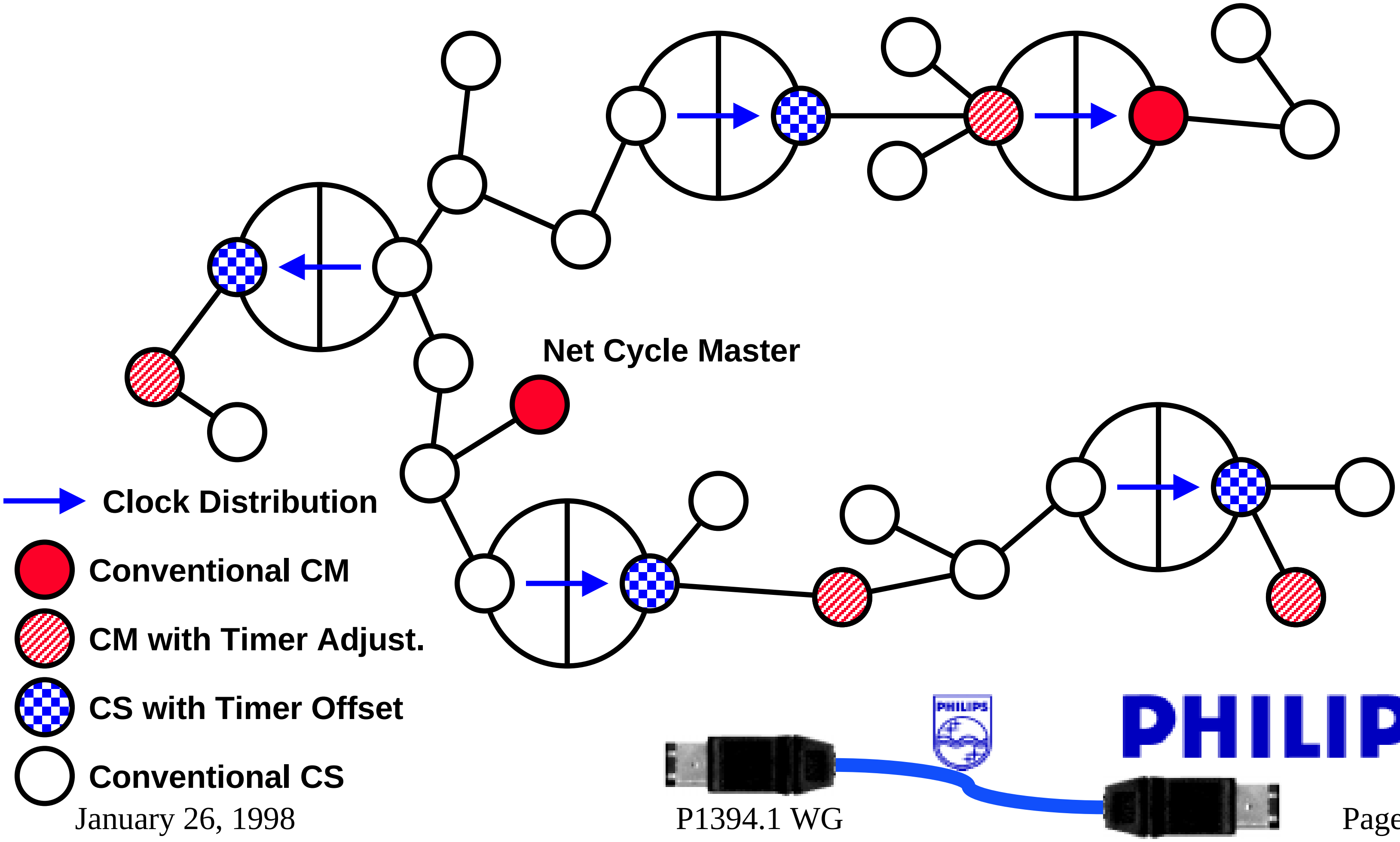
→ Clock Distribution

● Cycle Master

○ Cycle Slave

No More Limitation

- **Timer Offset/Adjustment Registers provide Full Flexibility of Cycle Master Location on each bus**



→ Clock Distribution

● Conventional CM

▨ CM with Timer Adjust.

▦ CS with Timer Offset

○ Conventional CS

Conclusion

- Bridge should implement Timer Offset Register in each portal
- 1394b Node should implement Timer Adjustment Register
- New silicon for generic purpose should implement both Timer Offset and Timer Adjustment registers
- Higher-layer protocol shall be defined
- More complete proposal (incl. Higher-layer protocol) will be presented at next P1394.1 meeting