1 Latch and Timing (Confidential)

1.1 @luk036

2021-11-10

class: middle, center

1.2 Introduction

1.2.1 Latch vs. Flip-Flop

.pull-left[

Latch:

- Level sensitive
- Timing analysis is difficult
- Lack of STA tools
- For low-power, high-speed design

] .pull-right[

Flip-Flop:

- Edge triggered
- Timing analysis is “easy”
- STA tools are available.
- Very common in any synchronous design.

]

https://en.wikipedia.org/wiki/Flip-flop_(electronics)

1.2.2 Timing constraints

- The clock arrival time is split into the earliest and the latest one, denoted by a_{f} and A_{f}, respectively.
  - $A_{\mathrm{f}} \geq a_{\mathrm{f}}$
- The clock departure time D_f and d_f are defined similarly.
- In addition to the setup- and hold-time constraints, there are propagation constraints:

$$\begin{aligned}
D_i &= \max(A_i, \phi_i + D_i) \\
d_i &= \max(a_i, \phi_i + d_i) \\
A_i &= \max_j [D_j + C^{j,i} + T_{\text{skew}}(j,i)] \\
a_i &= \min_j [d_j + c^{j,i} + T_{\text{skew}}(j,i)]
\end{aligned}$$

(Note: recurrence relation)

https://rd.springer.com/chapter/10.1007/978-0-387-71056-3_6

1.2.3 Max-Plus Algebra

A similar synchronous scheduling problem has been studied in for example, a rail system using the $(\max, +)$-algebra.

.pull-left[

Linear Algebra (+, ×, 0, 1):

- $a + b = b + a$
- $a + 0 = a$
- $1 \cdot a = a$
- $(a + b) \cdot c = a \cdot c + b \cdot c$] .pull-right[

Max-Plus (max, +, $-\infty$, 0):

- $\max(a, b) = \max(b, a)$
- $\max(a, -\infty) = a$
- $0 + a = a$
- $\max(a, b) + c = \max(a + c, b + c)$]

Unlike linear algebra, the max operation has no inverse (semi-ring).

https://en.wikipedia.org/wiki/Tropical_semiring#max-plus_algebra

1.2.4 Eigenvalue problem in $(\max, +)$ Algebra

- The recurrence relation can be expressed in terms of the Max-plus algebra:

  $$x = A \otimes x$$

  which is an eigen-problem.

https://www.degruyter.com/viewbooktoc/product/452553

1.3 Algorithms to solve the problem

- An obvious way to solve the problem is to use the Power method:
  - iterate recursively $x(k) = A \otimes x(k-1)$ until $x(k) = x(k-1)$.
  - The Power method is slow.
- **Surprisingly, the problem is equivalent to the maximum mean cycle problem, which can be solved efficiently by Howard's method.**

1.4 Timing Analysis

- Current approach 1: sort of like using the Power method to solve the eigenvalue problem, then check the setup- and hold-time violation.

- Power method is slow.
  - Power method cannot incorporate other design variables.
- Current approach 2: Treat the max operation as a non-linear function, then approximate the scheduling problem as mixed linear integer programming (MILP).
  - MILP is very slow.
  - MILP can incorporate any design variables.
- Approach 3: Howard's method
  - Howard's method is very fast.
  - The original method can only support one parameter.

1.5 Timing Optimization

- In a latch-based design, it was shown that the 50% duty cycle may not be optimal.
- Thus, both pulse width (W) and T_{CP} are design parameters.
- In this situation, the ellipsoid method can be used.

1.6 Advanced topics

- Multi-corner multi-mode
- Statistical timing analysis