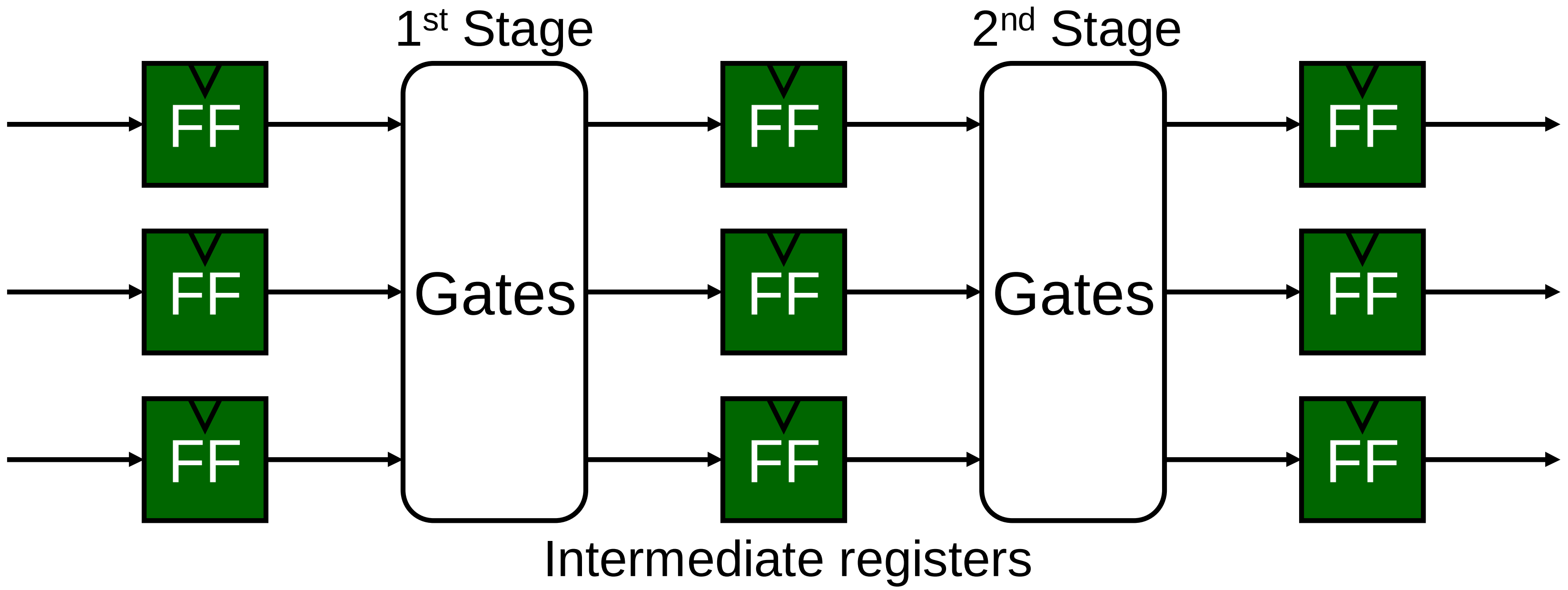
Low Area Pipelined Circuits by Multi-clock Cycle Paths and Clock Scheduling

Bakhtiar Affendi & Takahashi Atsushi
Comm. & Integrated Systems
Tokyo Institute of Technology

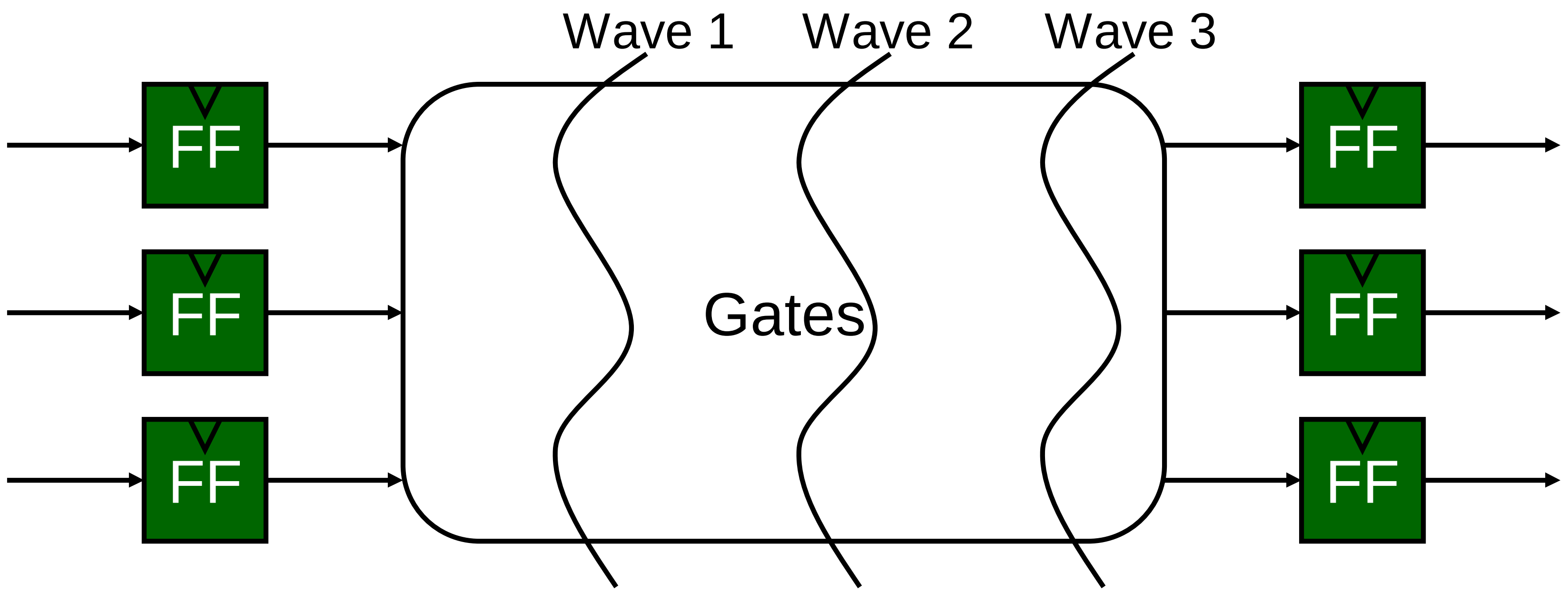
Pipelining

- A technique to shrink the clock period
- Circuit divided into number of stages.
- Intermediate registers inserted between stages.
- Problem of current pipelining method:-
    - Extra circuit area from the intermediate registers.
    - Increased the size of clock tree.



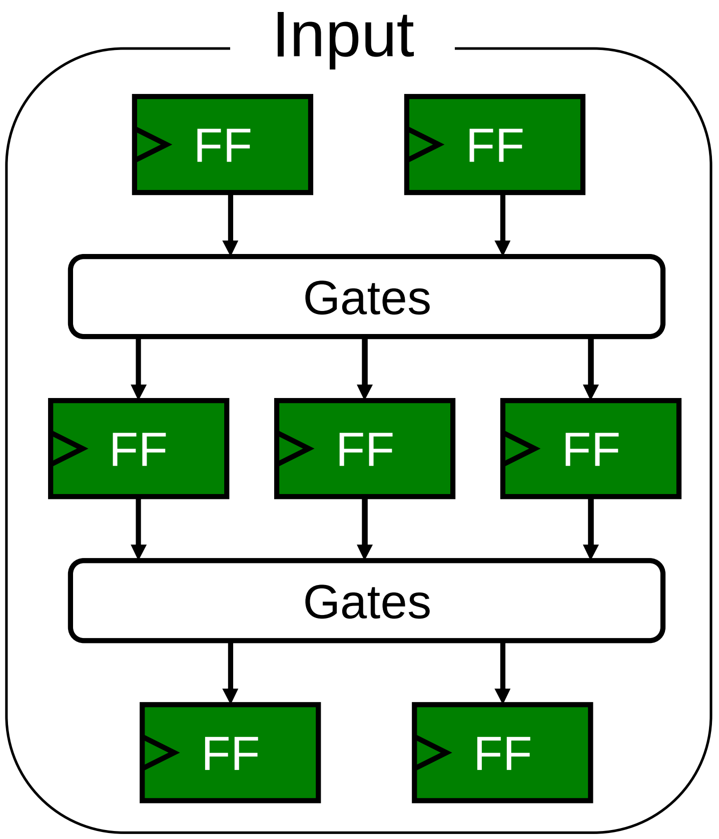
Wave Pipelining

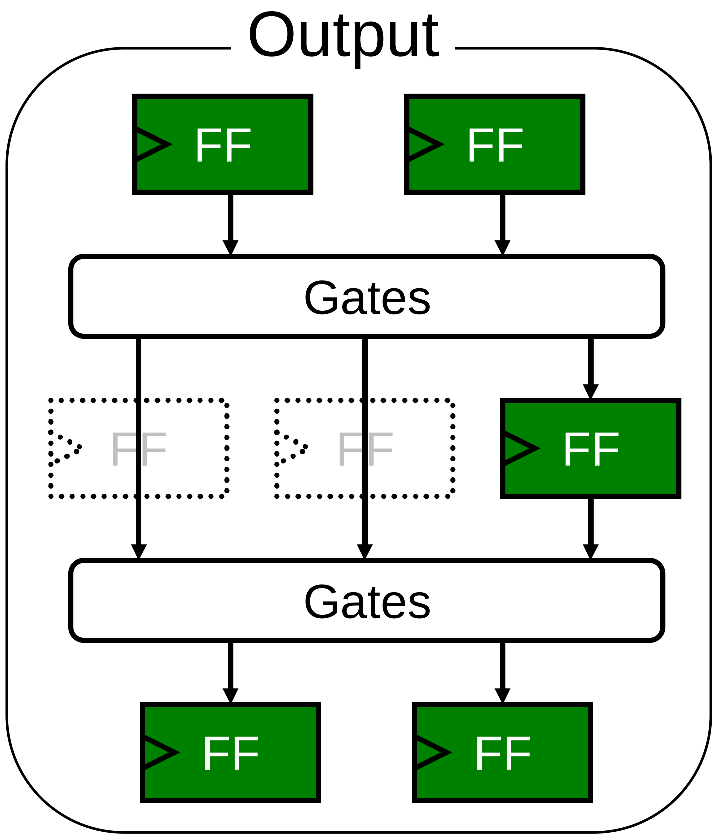
- Pipelining without the intermediate registers.
- Exist a number of waves of data in a stage.
- To avoid data collisions need delay balancing.
- Problem of Wave Pipelining:-
    - Delay balancing may increases circuit area.



Purpose

- Circuit with smaller area and works at target clock period range.
- Input :- Pipelined Circuit, clock period range.
- Objective:- Reduce the number of intermediate registers.

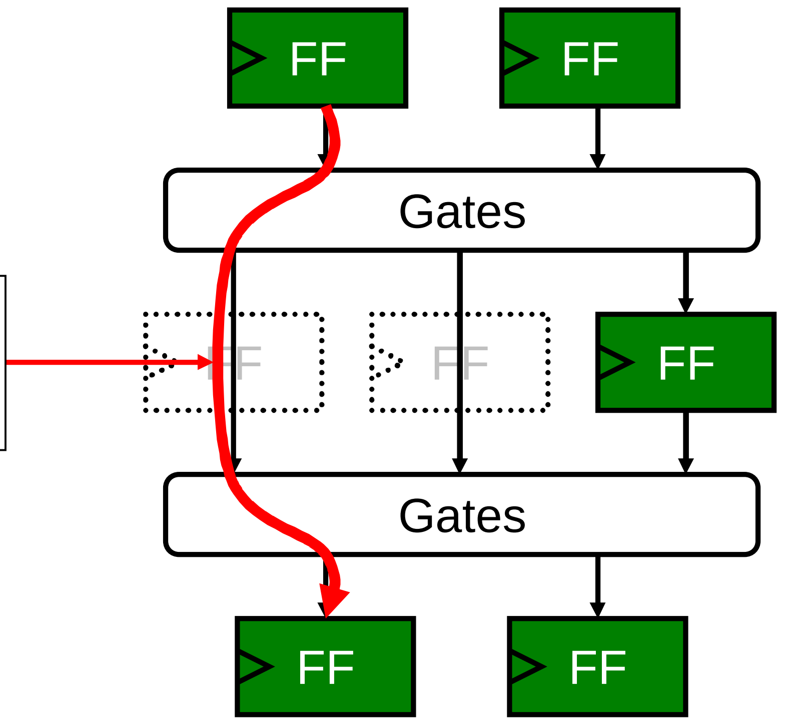




Method : Multi-clock Cycle Paths

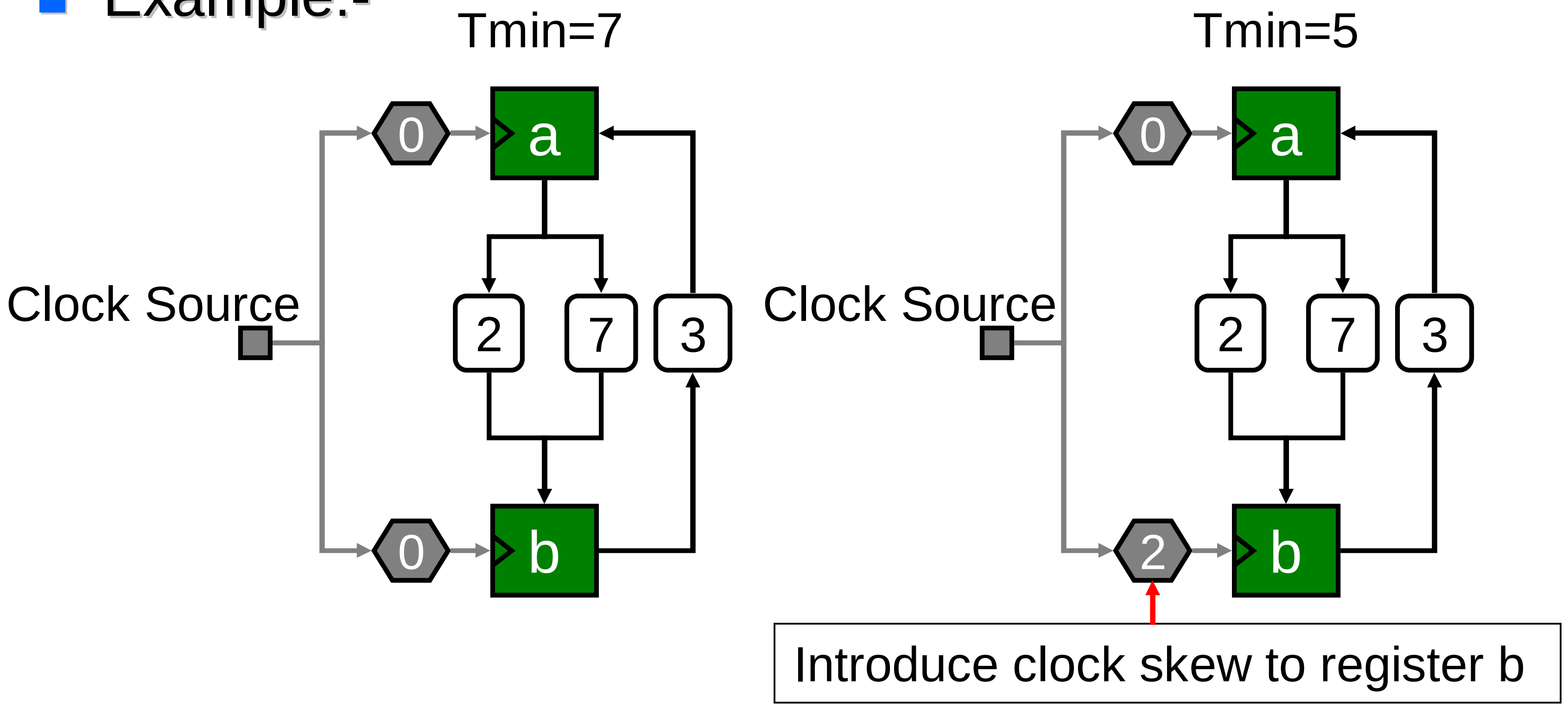
- Data transmission more than one clock period.
- Allows intermediate registers to be removed.
- Timing constraints must be satisfied.
  - Need delay balancing.

Example:-
Multi-clock cycle path



Method : Clock Scheduling

- Intentionally introduce clock skew of register
- Can relax timing constraints.
  - Can improve circuit performance.
- Example:-



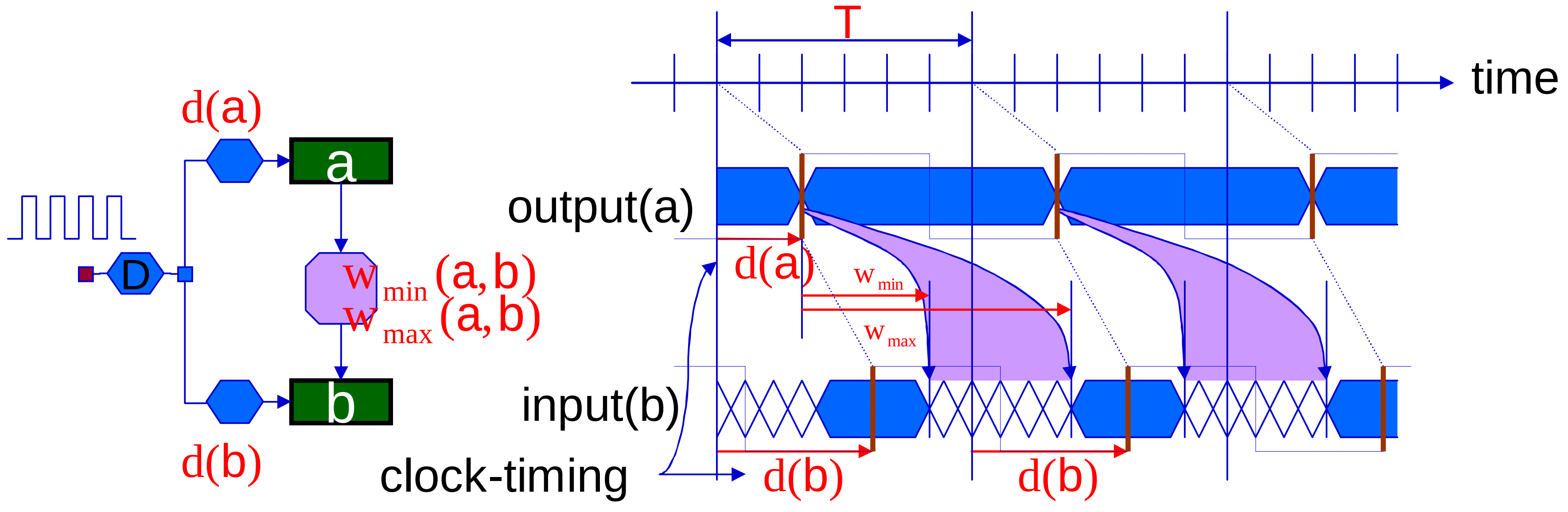
Clocking Constraints(1-clock-cycle)

- Setup Constraint (No zero-clocking)

$$d(a) + w_{max}(a,b) \le T + d(b)$$

- Hold Constraint (No double-clocking)

$$d(b) \le d(a) + w_{min}(a,b)$$



Clocking Constraints(Multi-clock-cycle)

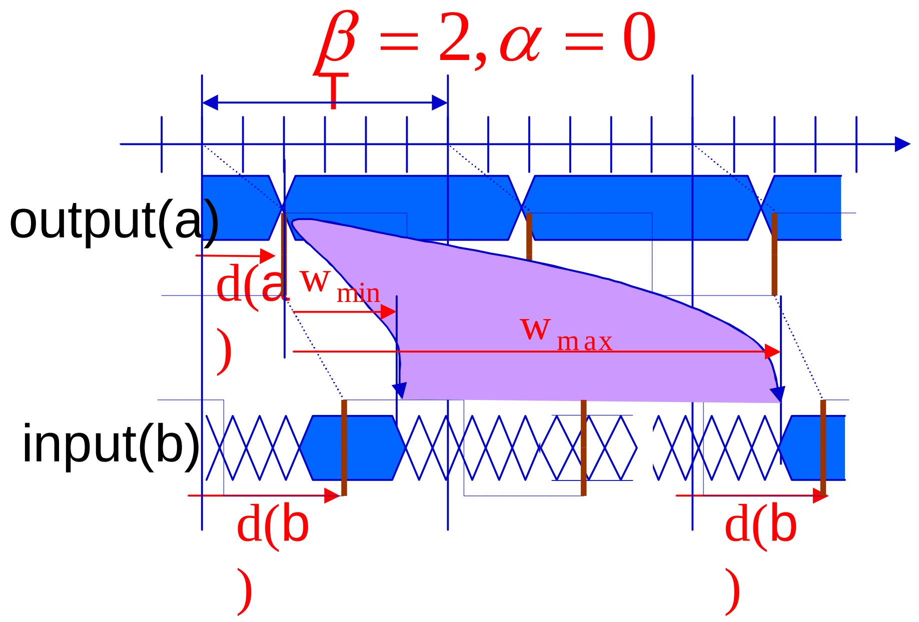
- Setup Constraint (No zero-clocking)

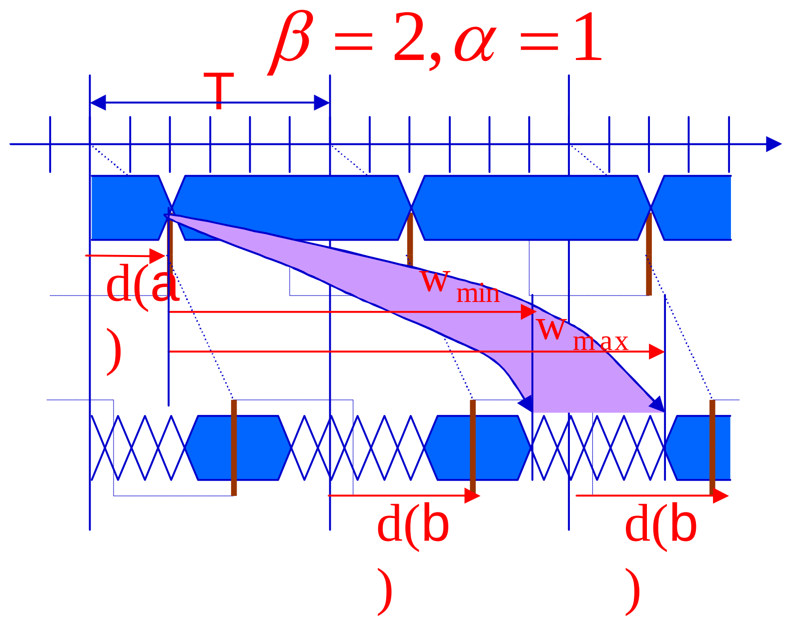
$$d(a) + w_{max}(a,b) \le \beta_{a,b} T + d(b)$$

- Hold Constraint (No double-clocking)

$$d(b) + \alpha_{a,b} T \le d(a) + w_{min}(a,b)$$

$$\beta_{a,b} > \alpha_{a,b} \ge 0$$





Bound of feasible clock period

- Setup Constraint (Decide lower bound, Tmin)

$$d(a) + w_{max}(a,b) \le \beta_{a,b}T + d(b)$$

$$\downarrow$$

$$\beta_{a,b}T \ge d(a) - d(b) + w_{max}(a,b)$$

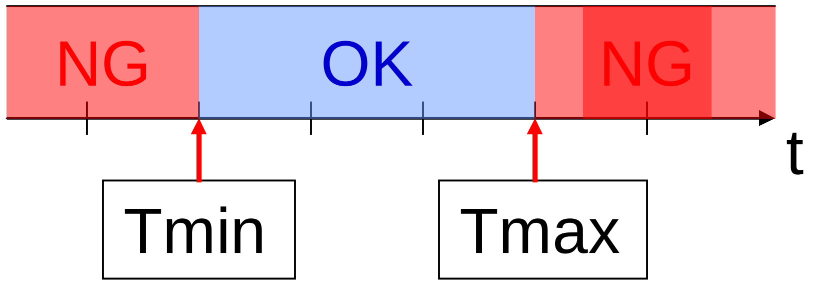
- Hold Constraint (Decide upper bound, Tmax)

$$d(b) + \alpha_{a,b}T \le d(a) + w_{min}(a,b)$$

$$\downarrow$$

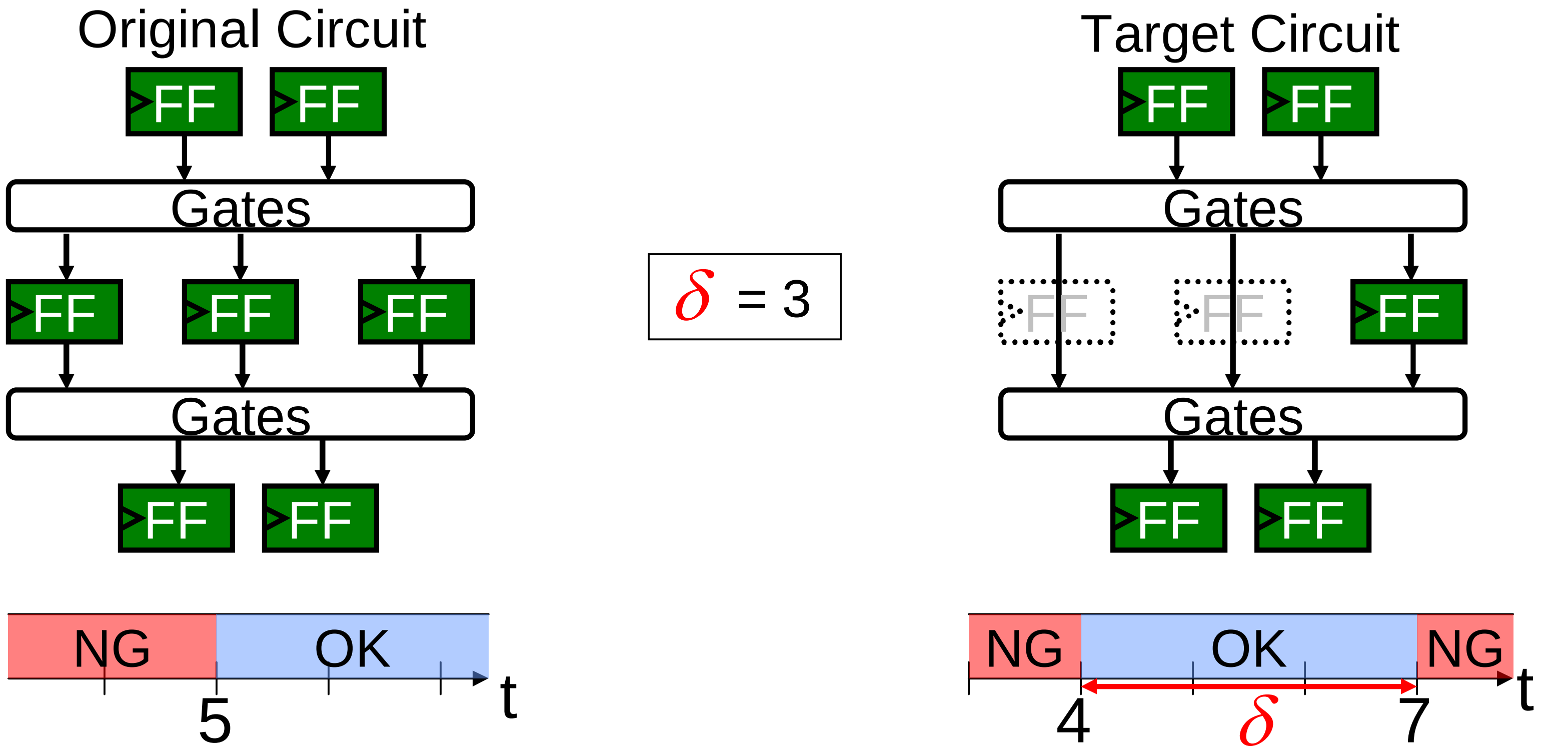
$$\alpha_{a,b}T \le d(a) - d(b) + w_{min}(a,b)$$

  – If $\alpha_{a,b} \ne 0$ exist an upper bound of clock period.



Target type of Circuit

- Reduce number of intermediate registers.
- Works at target clock period range δ.



- To reduce number of intermediate registers:-
  - Need to analyze a circuit with multi-clock cycle paths.
  - Need to know minimum feasible clock period of the circuit.

Clock Period Range δ

- Setup Constraint (Decide lower bound, Tmin)

$$\beta_{a,b} T_{min} \geq d(a) - d(b) + w_{max}(a,b)$$

- Hold Constraint (Decide upper bound, Tmax)

$$\alpha_{a,b} T_{max} \leq d(a) - d(b) + w_{min}(a,b)$$

$$\boxed{\delta = T_{max} - T_{min}}$$

$$\downarrow$$

$$\alpha_{a,b}(T_{min} + \delta) \leq d(a) - d(b) + w_{min}(a,b)$$

- If clock period range given, by using above constraints:-
  - Can get circuit works correctly between Tmin and (Tmin+ δ).

Minimum Clock Period with clock period range

- Problem formulation
  - Input: $w_{\max}(a,b), w_{\min}(a,b)$, clock period range δ.
  - Output: minimum clock period T.
  - Subject to: for each path between registers
    - $d(a) - d(b) \leq \beta T - w_{\max}(a,b)$
    - $d(b) - d(a) \leq w_{\min}(a,b) - \alpha\delta - \alpha T$

    β, α : Given constants

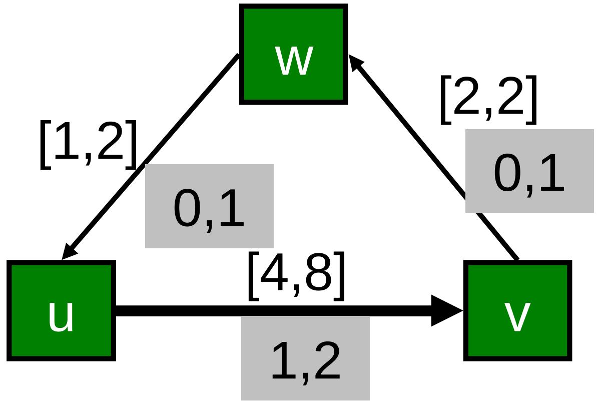
    $d(a)$: Clock input timing (variable)

  - Assumption
    - Arbitrary clock input timing is possible.
    - Clock-timing realization is independent of circuit. (given signal propagation delays are fixed)

Feasibility Check by Constraint Graph

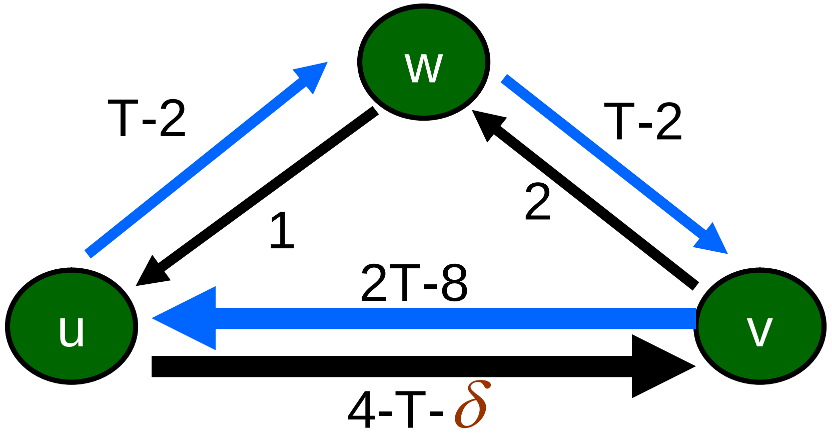
- Clock period T is feasible if and only if constraint graph has no negative weight cycle.
- Use shortest path algorithm to check there are negative cycle or not.

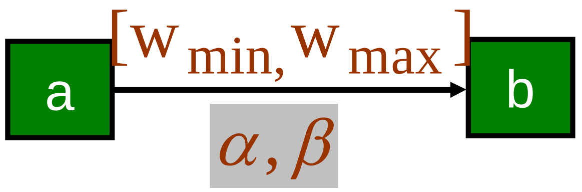
circuit graph

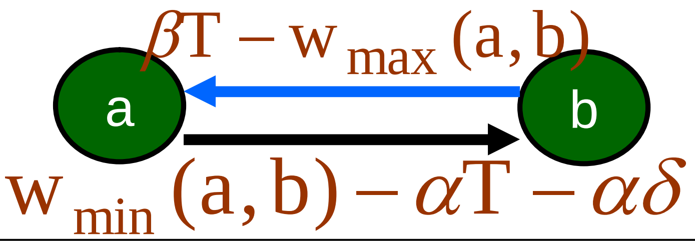


u-v is 2-clock cycle path

constraint graph





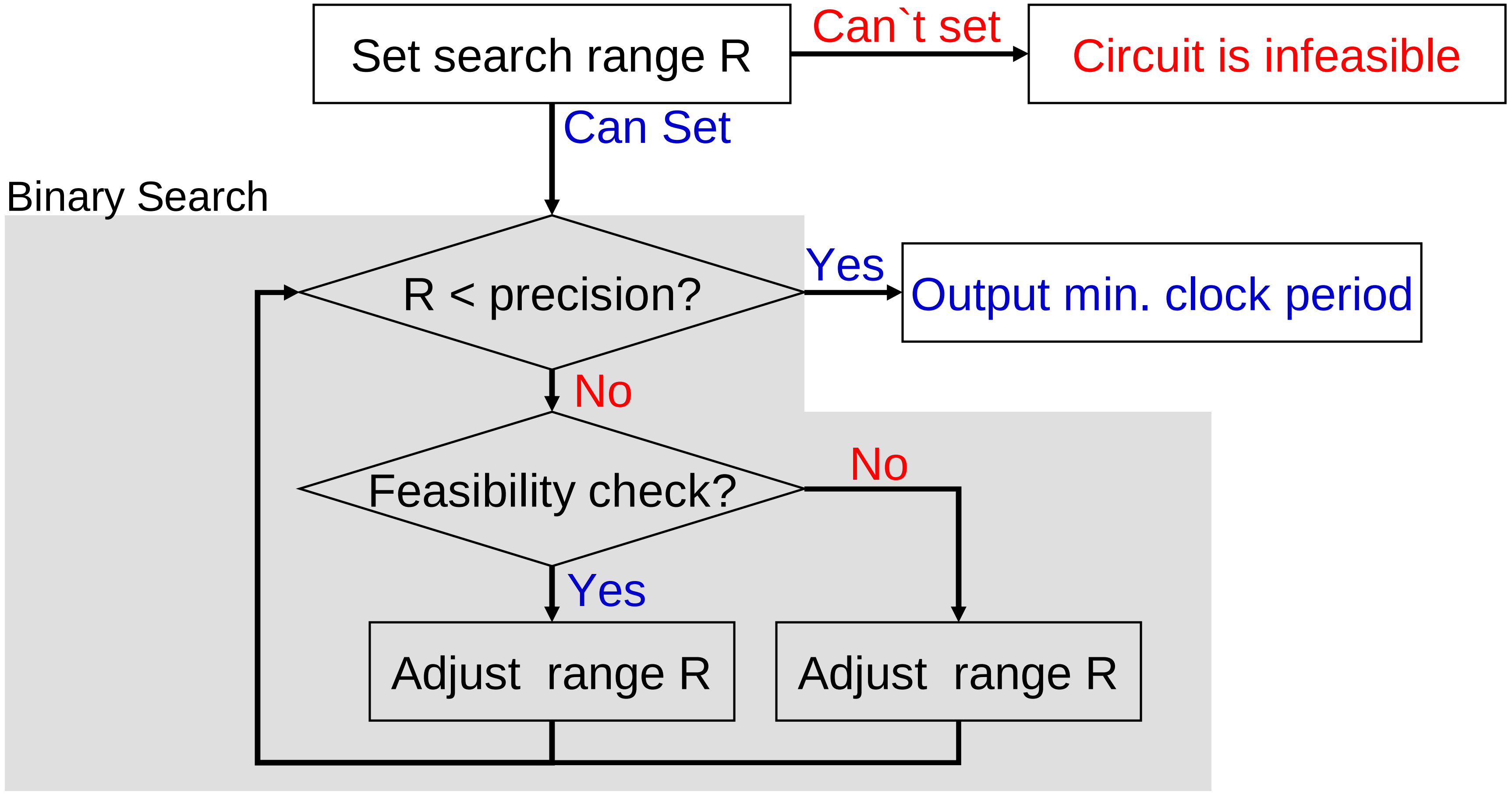


$$d(a) - d(b) \le \beta T - w_{max}(a,b)$$: Z-Edge (Setup)

$$d(b) - d(a) \le w_{min}(a,b) - \alpha T - \alpha\delta$$: D-Edge (Hold)

Original Algorithm [1] – Overview

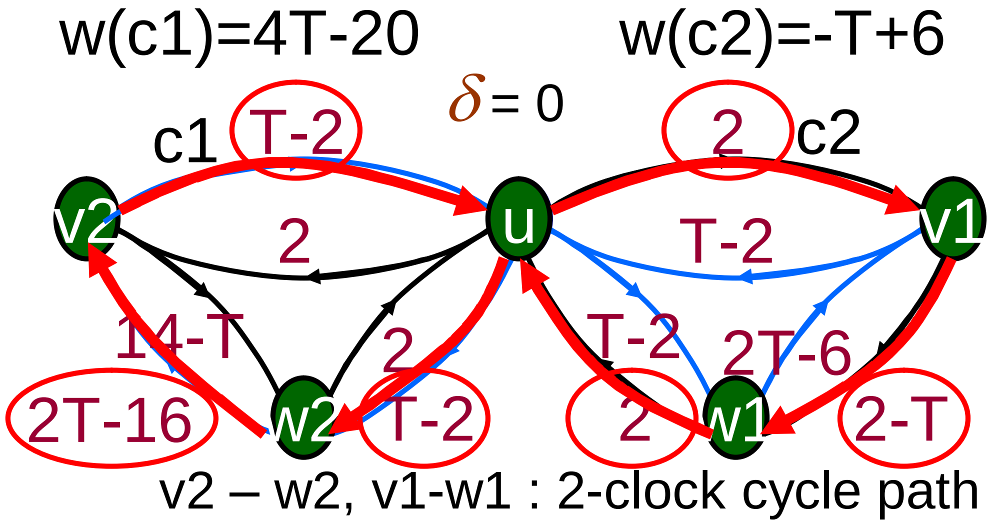
- Target circuit : Contains 1-clock cycle path only.

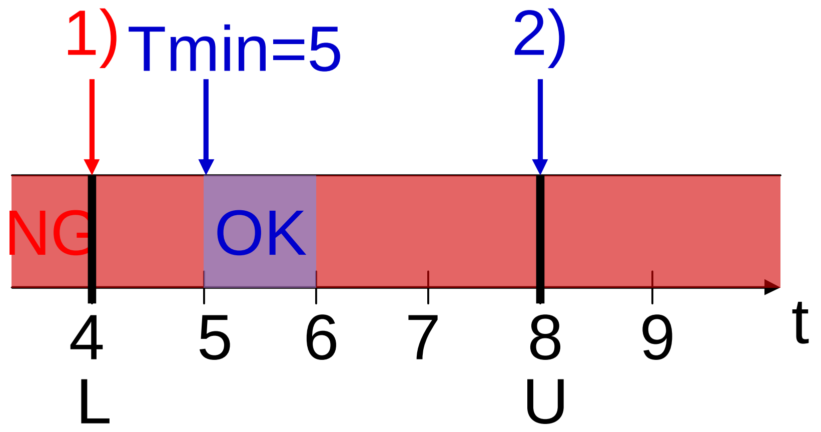


[1] : Practical Fast clock schedule design (Takahashi, 2005 18th karuizawa workshop)

Problem of Original Algorithm

- Apply original algorithm to circuit contains multi-clock cycle path.



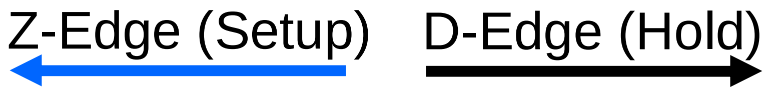


Searching Range (L=4,U=8)

1) (L=4,U=8) Check L :-
- T=4,w(c1)= 4T-20= -4 c1 is negative

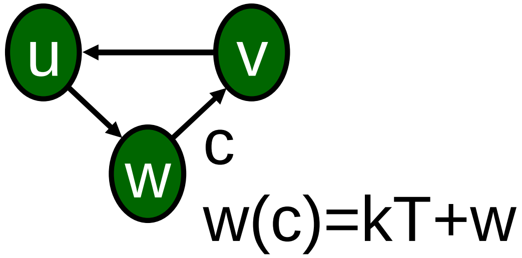
2) (L=4,U=8) Check U :-
- T=8,w(c2)= -T+6= -2 c2 is negative

Algorithm assumes that the circuit is infeasible at any clock period.

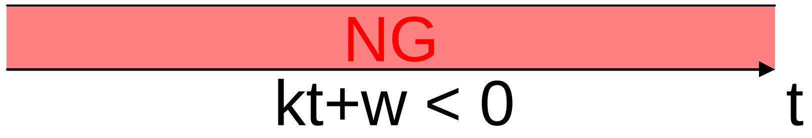


Definition for type of cycle

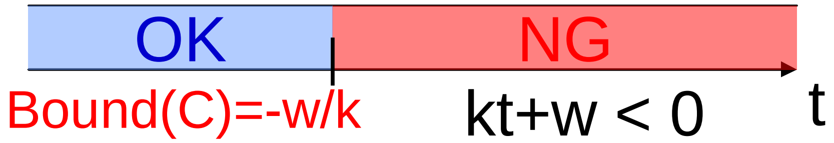
- Cycle weight of C, w(C) = kT+w.
  k,w: constants
  T: clock period



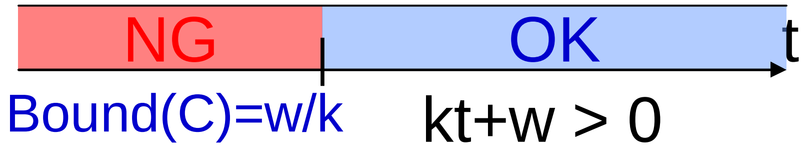
- k=0, C is 0-type.
  - If w < 0, circuit is infeasible for any T.



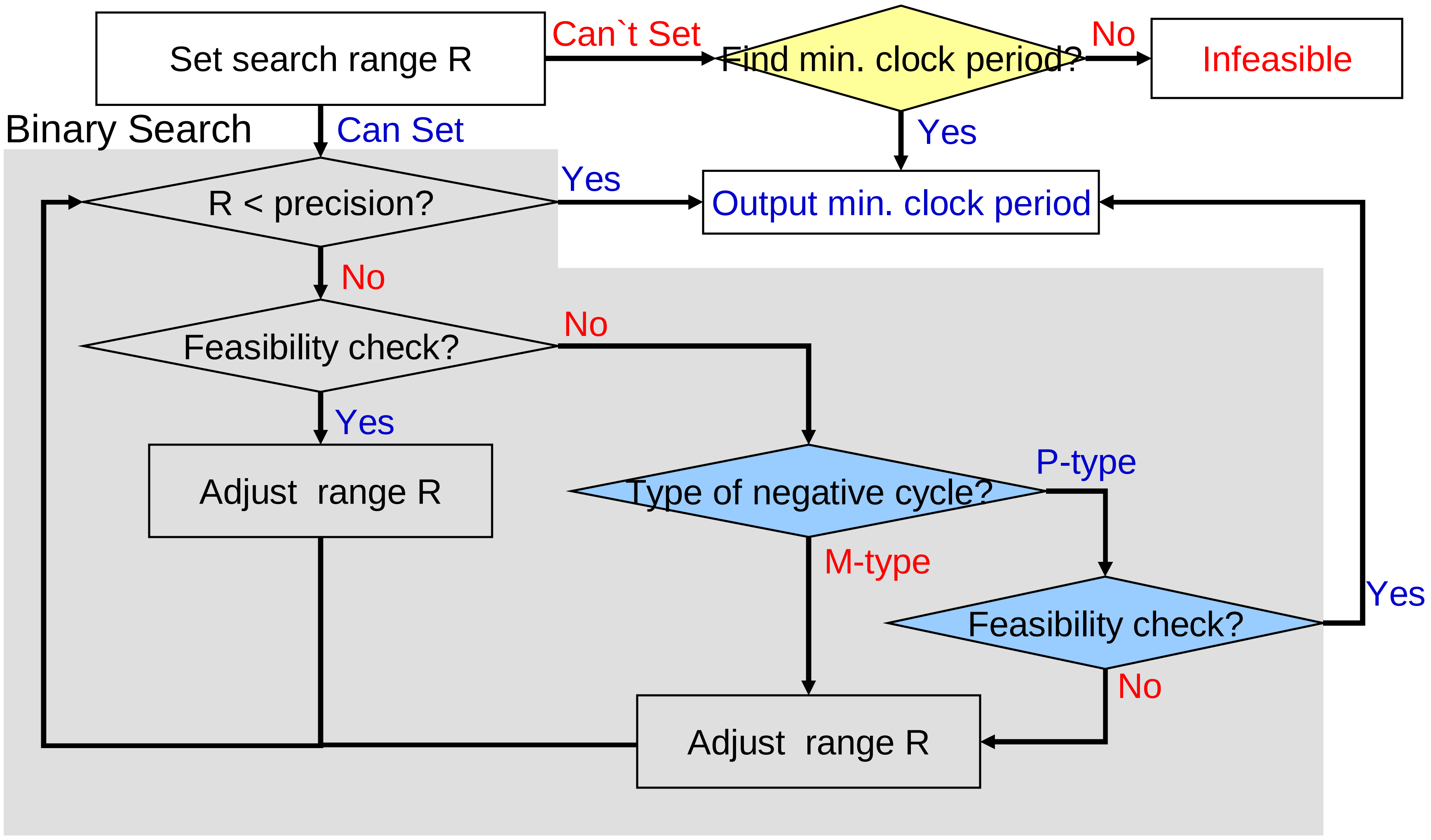
  - If w$\geq$ 0, no problem.
- k<0, C is M-type.



- k>0, C is P-type.

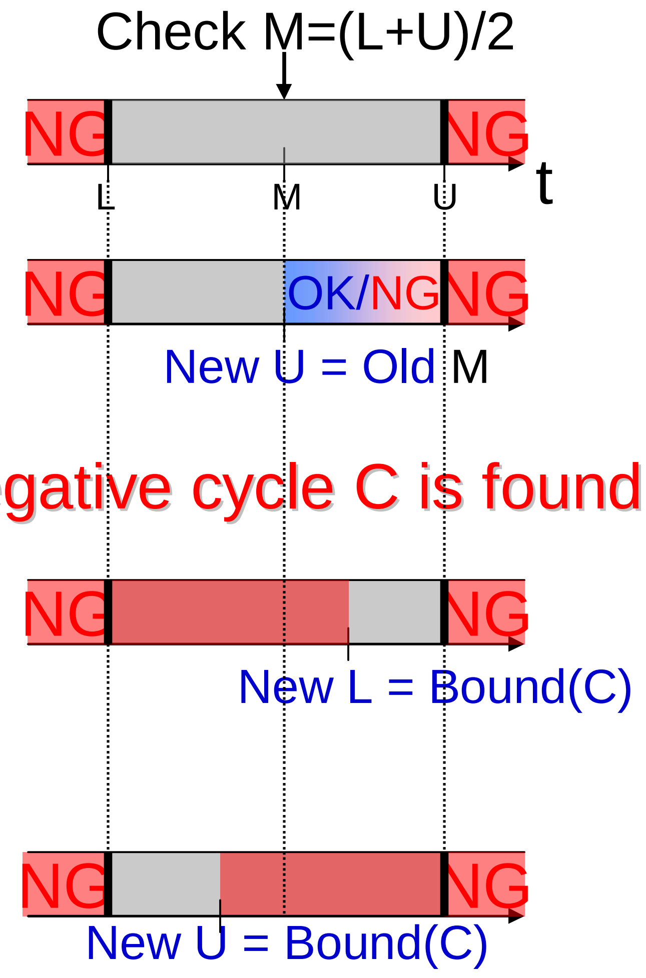


Proposed Algorithm – Overview

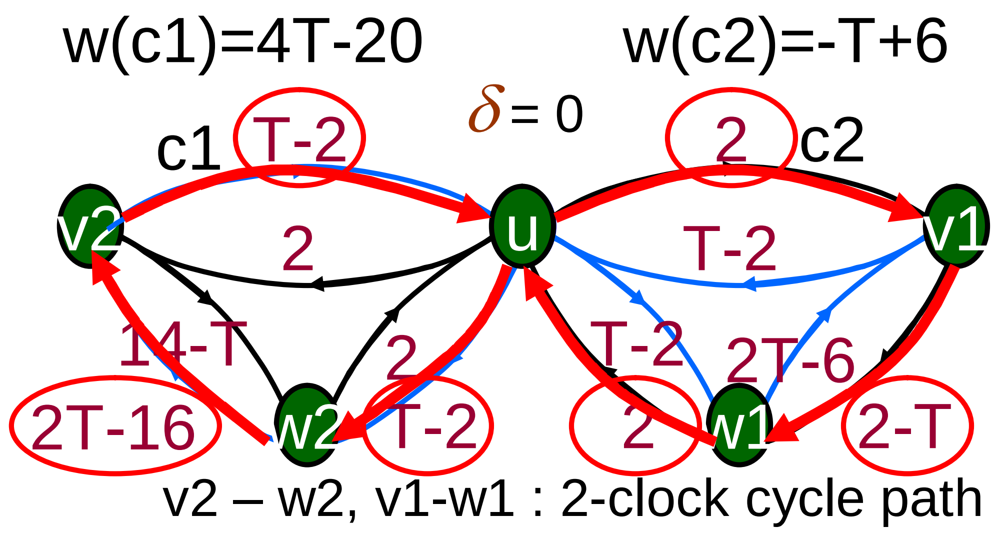


Proposed Algorithm: Binary Search

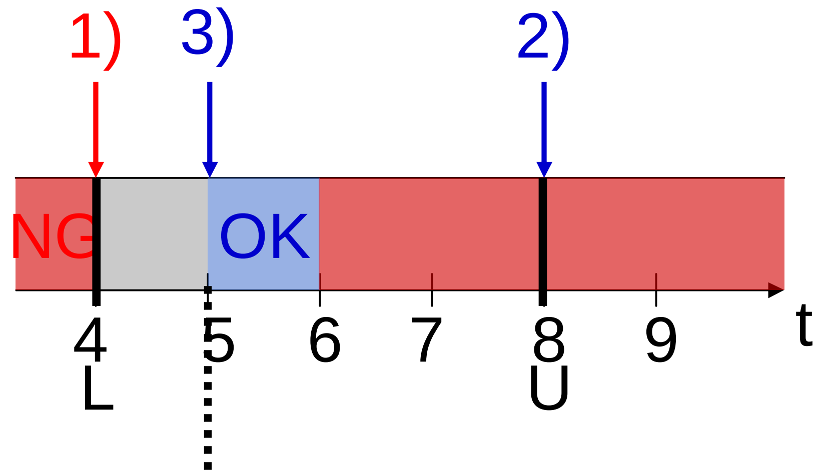
- Feasibility check
  - Feasible:-
  - Infeasible, a negative cycle C is found:-
    - C is P-type
    - C is M-type



Proposed Algorithm: Example



v2 – w2, v1-w1 : 2-clock cycle path



Tmin = 5, Precision=1

Searching Range (L=4,U=8)

1) (L=4,U=8) Check L :-

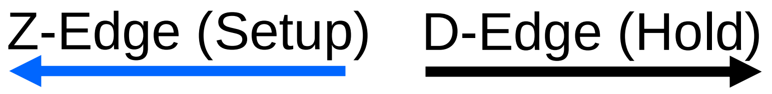
- T=4,w(c1)= 4T-20= -4
  c1 is negative,P-type

2) (L=4,U=8) Check U :-

- T=8,w(c2)= -T+6= -2
  c2 is negative,M-type
  Bound(c2)=6 = new U

3) (L=4,U=6)
Check M=(4+6)/2=5:-

- T=5,no negative cycle



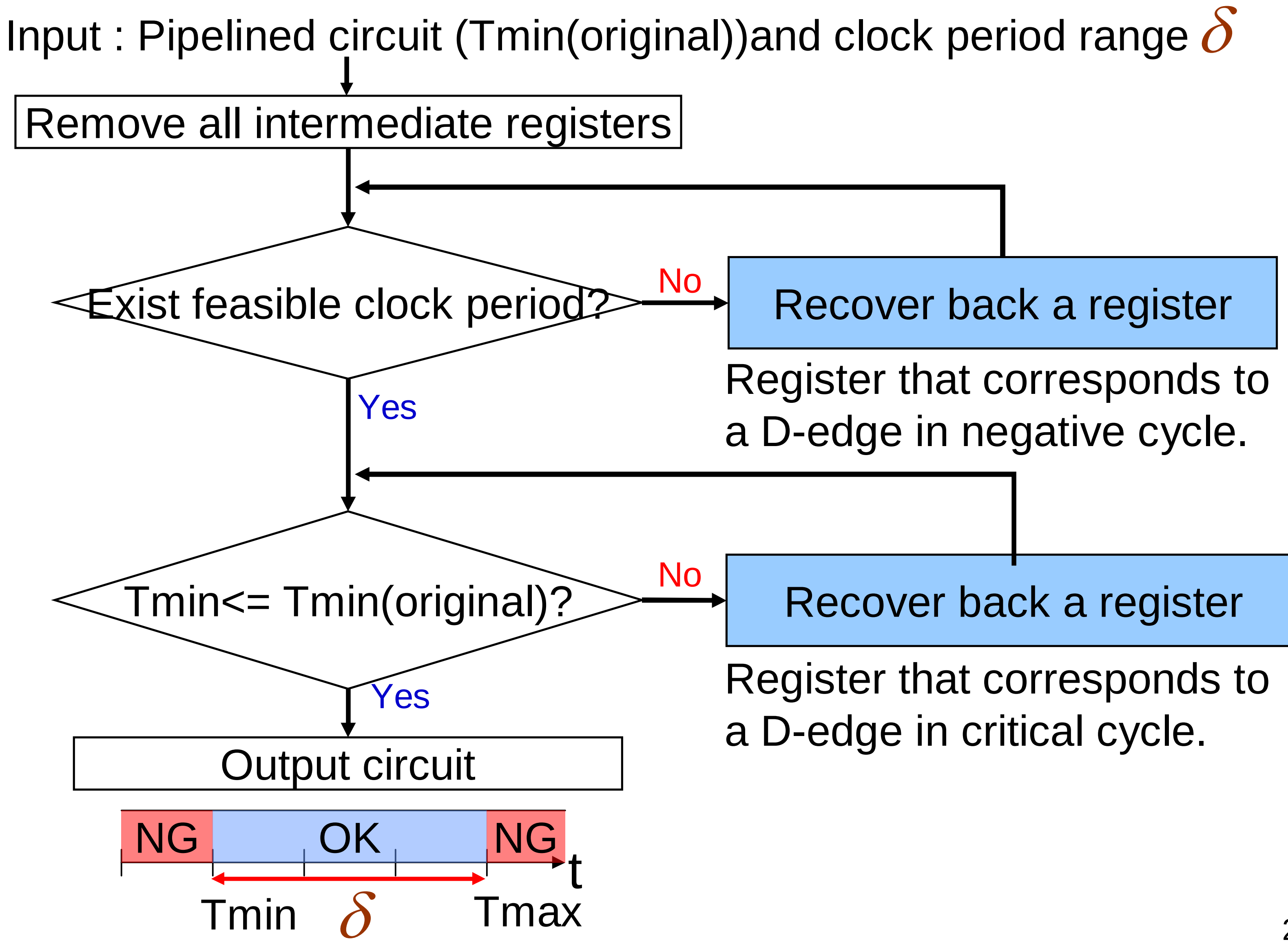
Experiment Results

		Proposed		Original	
*δ [ps]	circuit	Tmin [ps]	Tmax [ps]	Tmin [ps]	Tmax [ps]
0	Add1	2407	2407	NA	NA
150	Add1	2560	2710	NA	NA
0	Add2	2231	2231	2231	2231
250	Add2	2560	2810	NA	NA

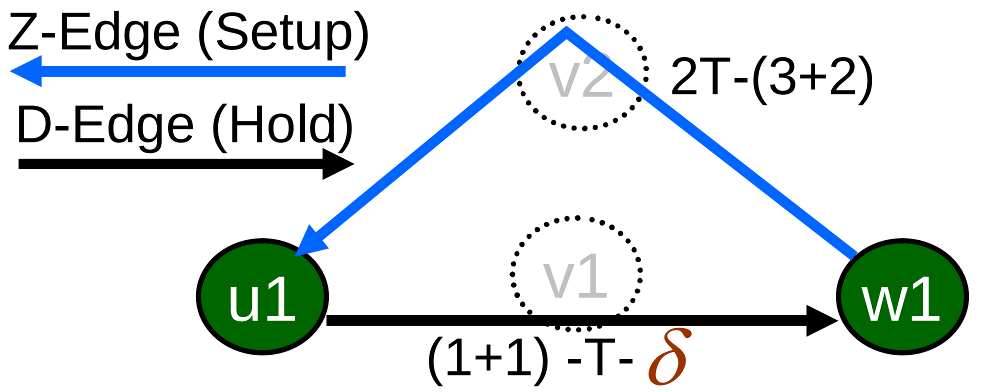
*δ is target clock period range (Input)

- Proposed algorithm can find minimum feasible clock period of a circuit with multi-clock cycle paths.

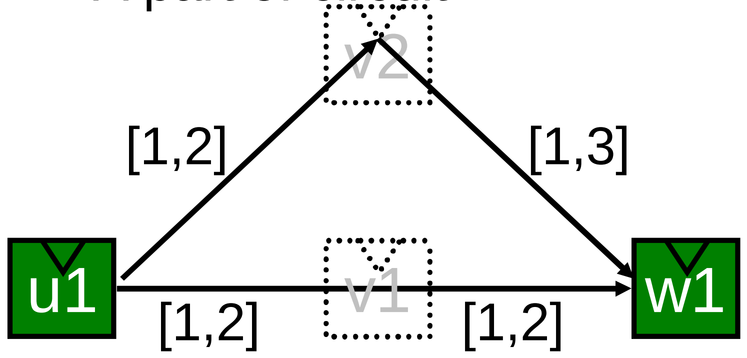
Reduction on the number of registers


Input : Pipelined circuit (Tmin(original))and clock period range δ
Remove all intermediate registers
Exist feasible clock period?
No
Recover back a register
Register that corresponds to a D-edge in negative cycle.
Yes
Tmin<= Tmin(original)?
No
Recover back a register
Register that corresponds to a D-edge in critical cycle.
Yes
Output circuit
NG
OK
NG
t
Tmin
δ
Tmax

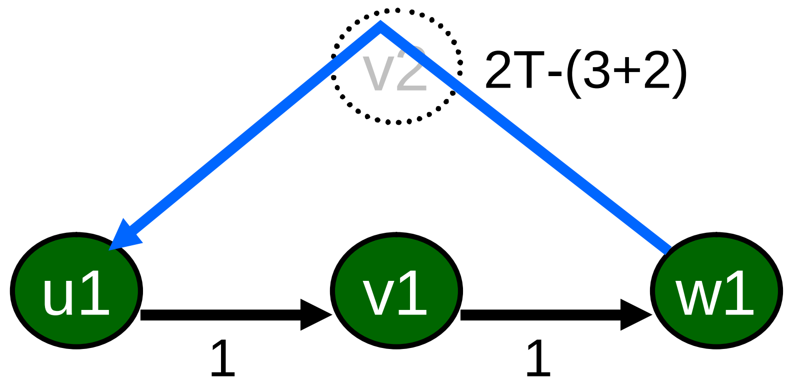
Why recover back a register corresponds to D-Edge?

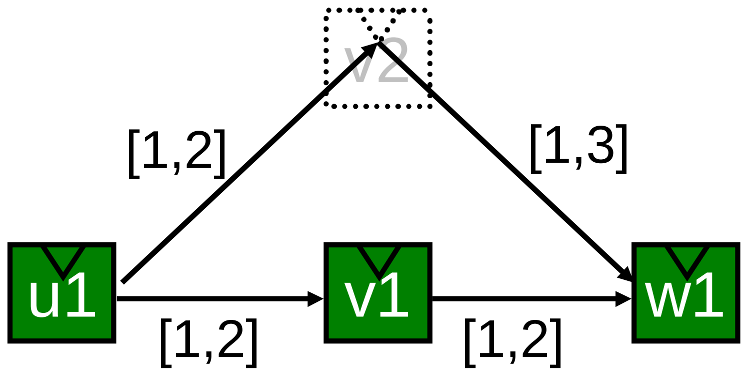


A part of circuit



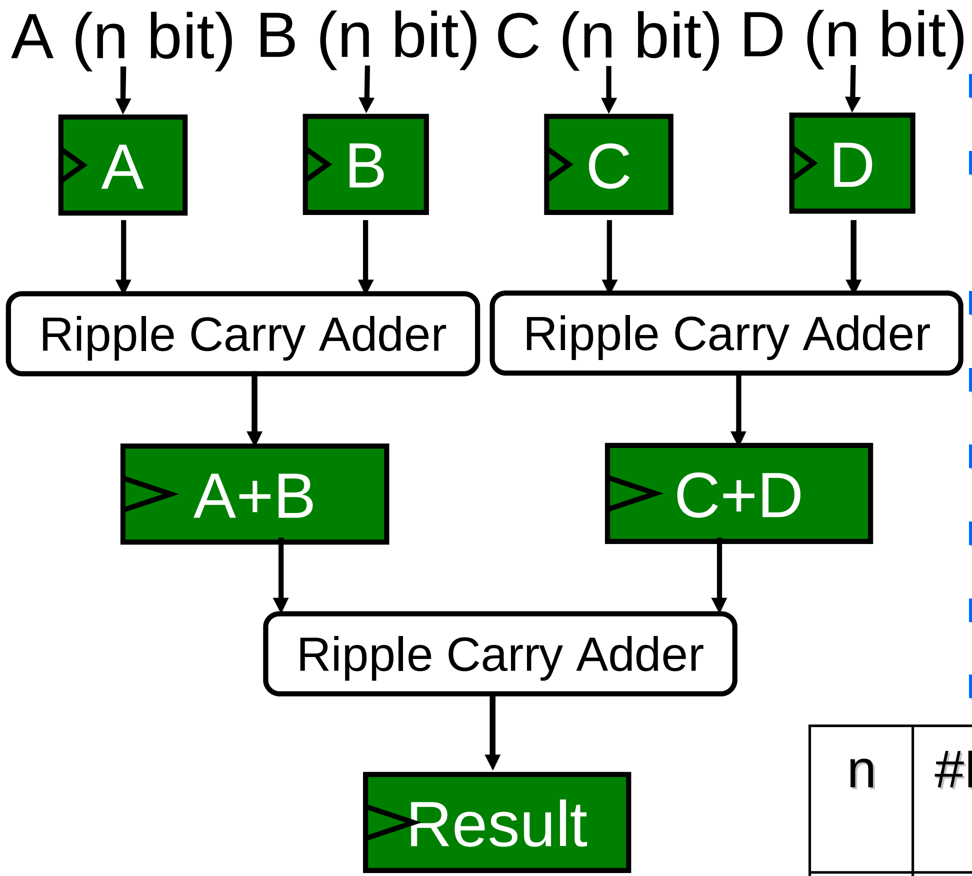
T=2, δ=1 w(u1,w1,u1) = 2T-(3+2) +(1+1)-T- δ= -2 : negative cycle due to minimum delay path u1,v1,w1.





T=2, δ=1 Recover back register v1
w(u1,v1,w1,u1) = 2T-(3+2) +1+1 = 1

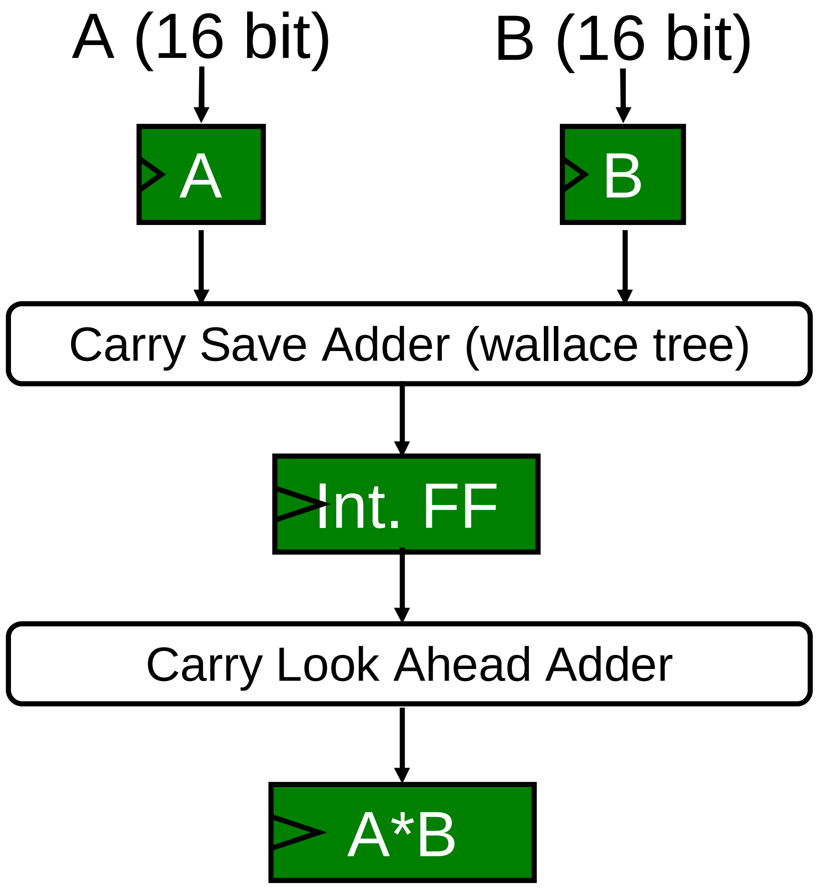
Experiments : Input Circuit (Pipelined Adder)



- 2-Stage.
- Add four n-bit numbers. (n=4,8,16).
- 1st stage : A+B and C+D.
- 2nd stage : A+B+C+D.
- Each adder : Ripple Carry Type.
- Process Library ROHM 0.35um.
- Scheduled I/O pins and registers.
- Circuit Statistics:-

n	#FF	1st Stage (ps)		2nd Stage (ps)	
4	32	588	2454	588	2840
8	60	598	4079	598	4474
16	116	598	7239	598	7634

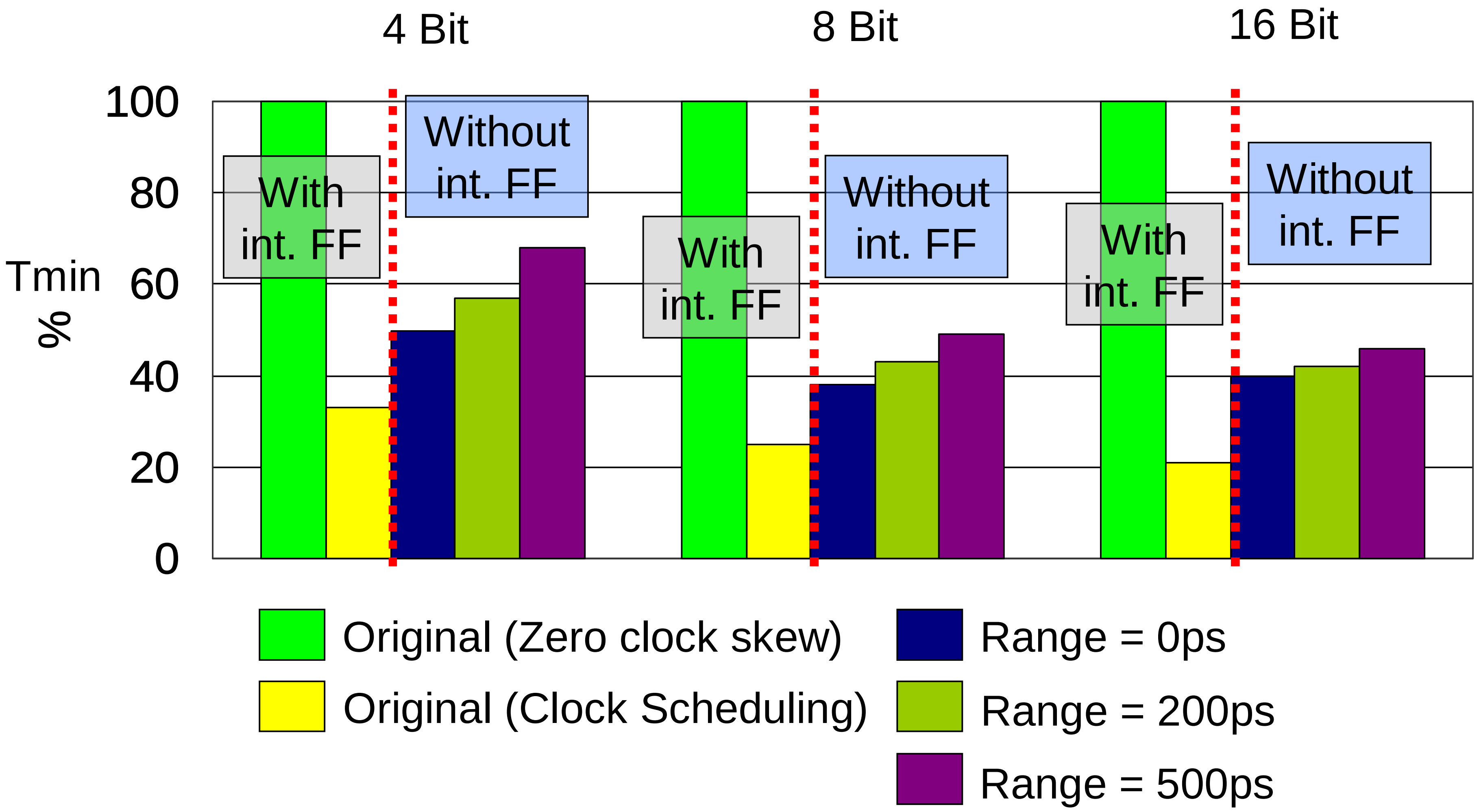
Experiments : Input Circuit (Pipelined Multiplier)



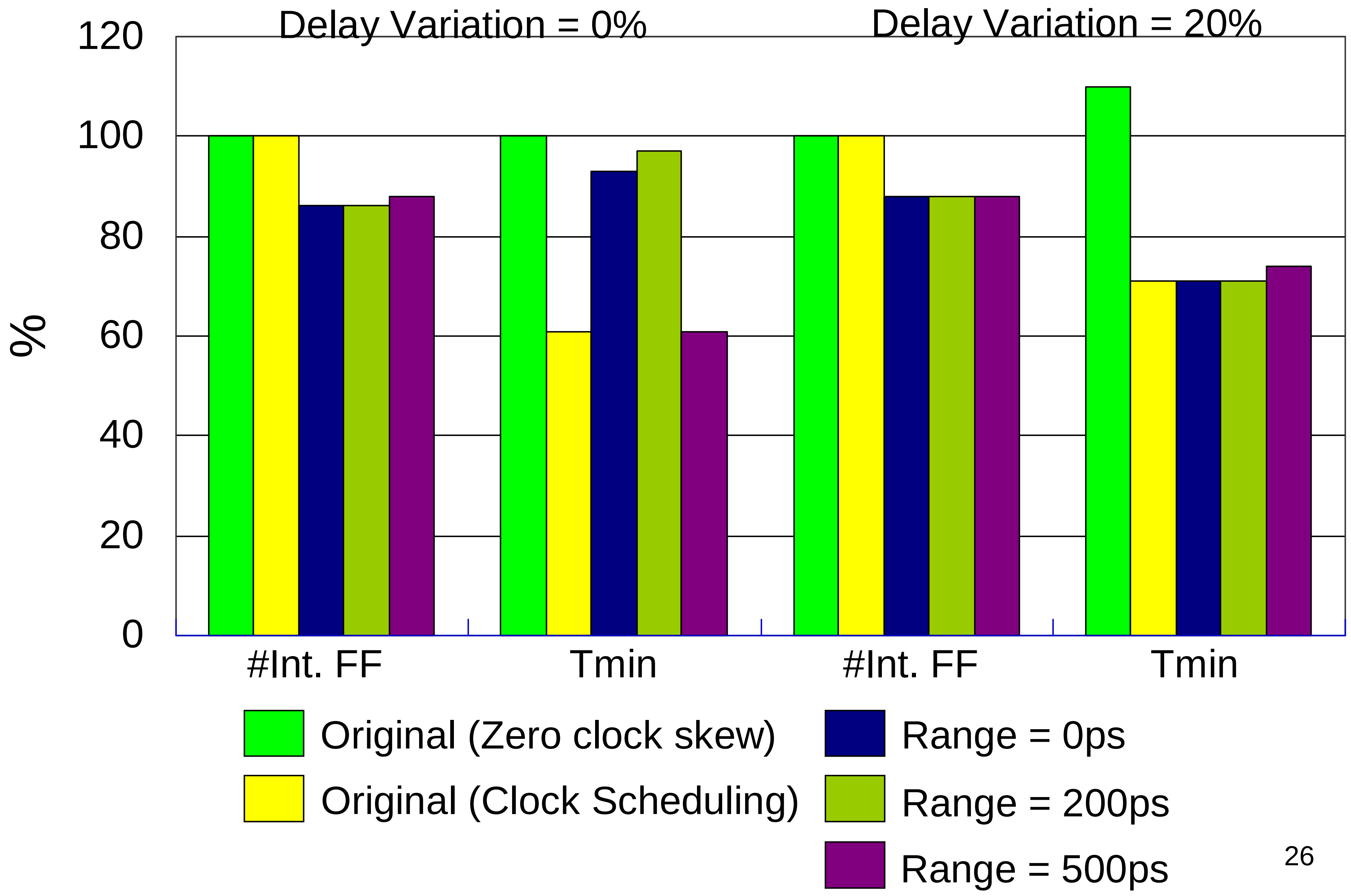
- 2-Stage.
- Multiply two 16-bit numbers.
- 1st stage : Carry Save adder with wallace tree structure.
- 2nd stage : Carry look ahead adder.
- Process Library ROHM 0.35um.
- Scheduled I/O pins and registers.
- Circuit Statistic:-

#FF	1st Stage (ps)		2nd Stage (ps)	
120	757	5075	373	4050

Experiments Results : Pipelined Adder



Experiments Results : Pipelined Multiplier



Conclusions

- Conclusion:-
  - Number of intermediate registers can be reduced by multi-clock cycle paths and clock scheduling.
- Future Works:-
  - Proposed algorithm only insert intermediate registers to satisfy timing constraints.
  - Consider delay balancing together with intermediate register insertion.