



ASAT

PART II

인공지능 시스템 반도체 응용 전문가

AI System Semiconductor Application Specialist Academy & Certificate Test

디지털 회로 가속 설계 방법

명지대학교 박상윤 교수



서강대학교



서울과학기술대학교



아주대학교



국립한밭대학교



한국생산성본부

CONTENTS

4.1. 디지털 회로의 동작과 타이밍

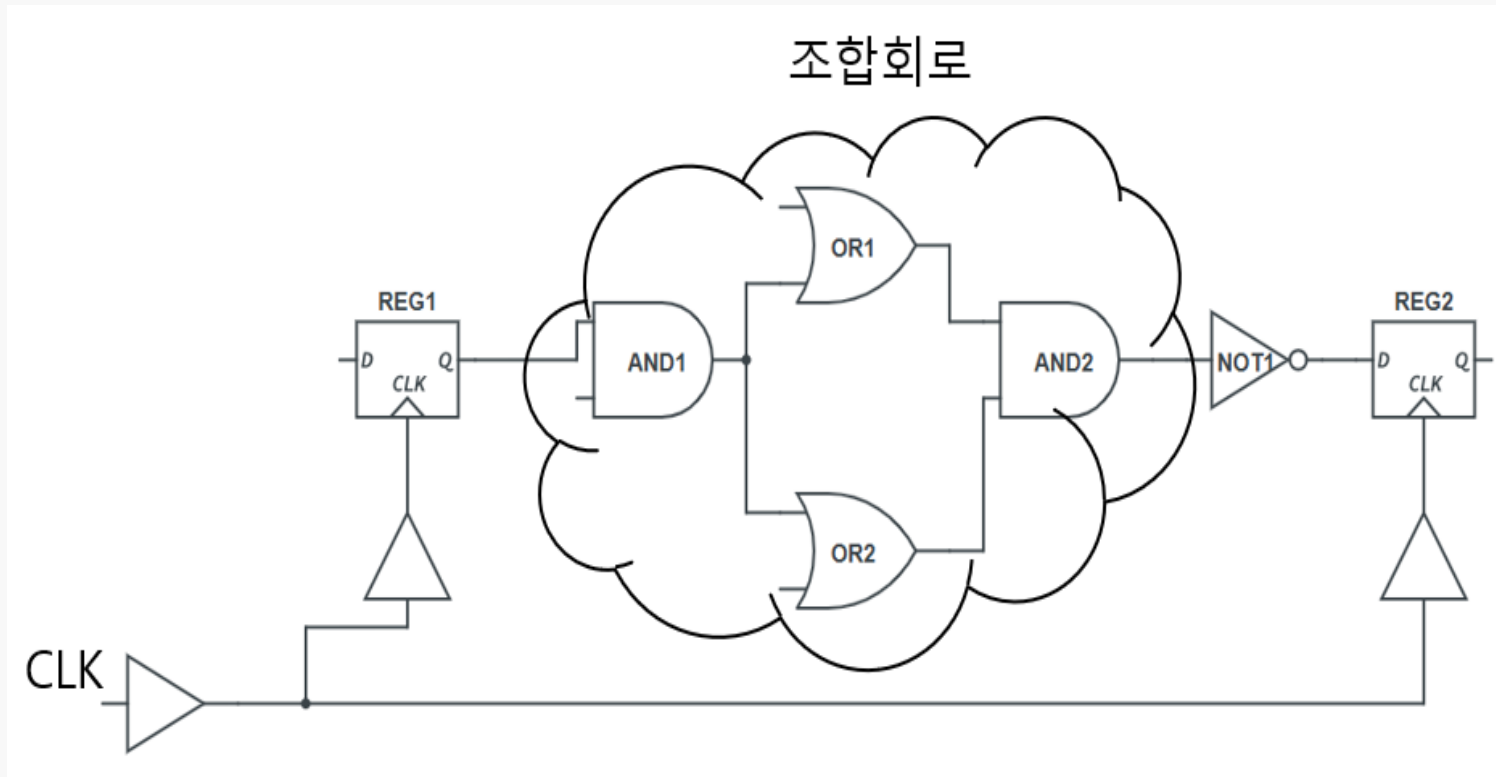
4.2. 디지털 회로의 동적 규율

4.3. 디지털 회로의 가속 설계 방법

4.4. 유한 임펄스 응답 필터의 가속 설계 방법

4.1. 디지털 회로의 동작과 타이밍

◆ 디지털 회로의 구성



4.1. 디지털 회로의 동작과 타이밍

◆ 조합회로

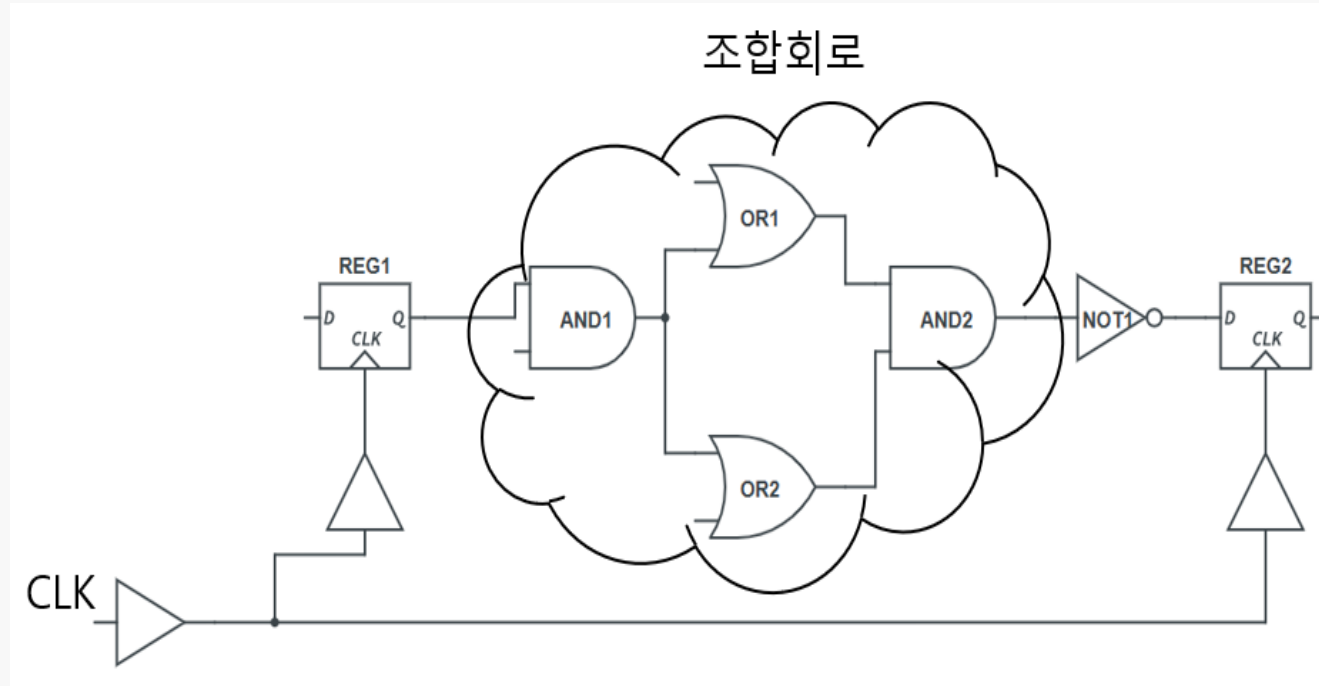
- ❖ 현재의 입력에만 의존하여 출력을 생성하는 회로
- ❖ 입력신호가 들어오면 즉각적으로 연산을 시작
- ❖ 논리 게이트를 포함하며, 다양한 연산이나 논리 처리를 담당
- ❖ 각각의 게이트에서 발생하는 지연시간 만큼 총 지연 시간이 누적됨
- ❖ 덧셈기, 인코더, 디코더, 멀티플렉서 등

◆ Flip-Flop, Register

- ❖ Flip-Flop 하나는 1비트 저장,
- ❖ 레지스터는 N개의 Flip-Flop의 묶음으로서, N비트 저장
- ❖ 클럭의 에지에서 입력을 샘플링하여 저장하고, 출력으로 내보내는 역할
- ❖ 신호의 스케줄링과 임시 저장소 역할 수행

4.1. 디지털 회로의 동작과 타이밍

◆ 타이밍 조건이란?

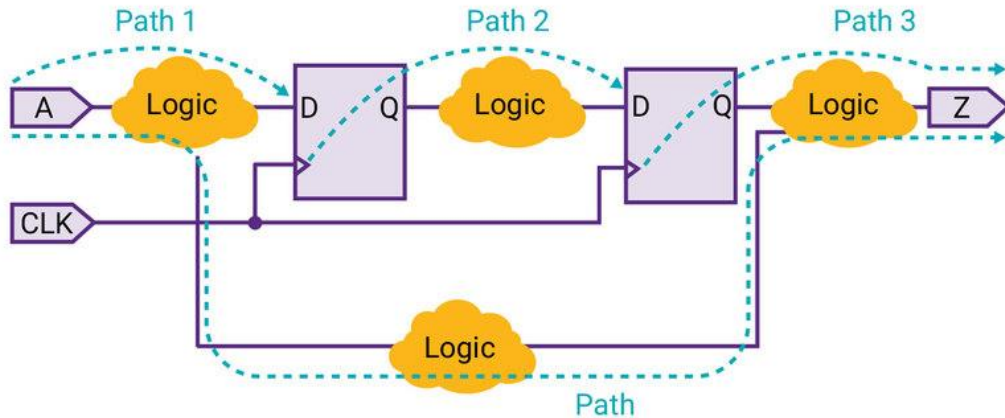


- ✓ 레지스터 1의 클럭 (CLK) 신호의 에지가 발생한 후, 레지스터 2의 입력단에 도착할 때까지 100ns가 걸린다면, CLK의 주기는 최소 100ns 이상이어야 함
- ✓ 타이밍 조건 (timing constraint)이란, 회로가 올바르게 동작하기 위해 신호가 레지스터와 조합회로 사이에서 지켜야 하는 최소 및 최대 시간 제약

4.1. 디지털 회로의 동작과 타이밍

SYNOPSYS®

Timing paths

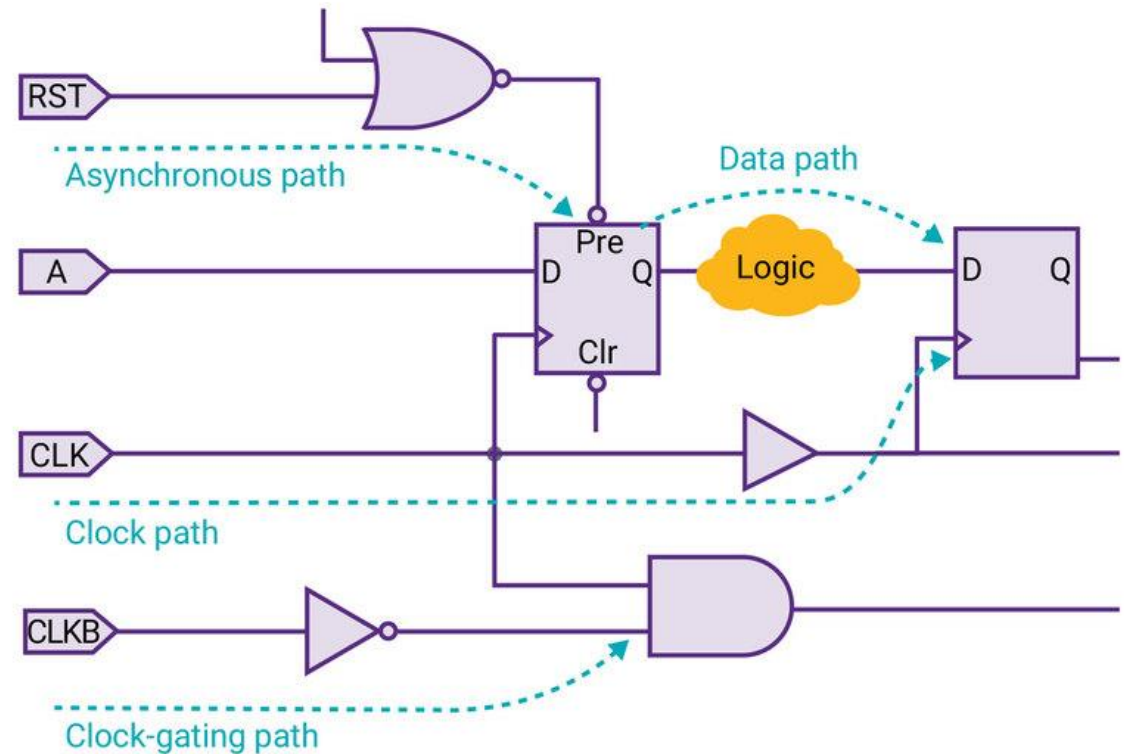


In the example, each logic cloud represents a combinational logic network. Each path starts at a data launch point, passes through some combinational logic, and ends at a data capture point.

Path	Startpoint	Endpoint
Path 1	Input port	Data input of a sequential element
Path 2	Clock pin of a sequential element	Data input of a sequential element
Path 3	Clock pin of a sequential element	Output port
Path 4	Input port	Output port

SYNOPSYS®

Types of paths considered for timing analysis



4.1. 디지털 회로의 동작과 타이밍

◆ 정적 시간 분석 (Static Timing Analysis, STA)

- ❖ 디지털 회로 내에 모든 가능한 경로에서 타이밍 위반이 있는지 확인하여 성능을 검증하는 단계
- ❖ 회로의 모든 경로에 대해 입력 신호와는 무관하게 지연 시간을 계산하여, 타이밍 제약을 만족하는지 확인
- ❖ 시뮬레이션 없이 수학적 분석을 통해 이루어짐

◆ 동적 시간 분석 (Dynamic Timing Analysis)

- ❖ 실제 입력 신호를 사용하여 회로를 시뮬레이션하고, 특정 입력 조건에서의 타이밍 성능을 확인하는 방식
- ❖ 입력 신호에 따라 다르게 동작할 수 있는 다양한 시나리오를 검증할 수 있지만, 정적 시간 분석보다 분석에 더 많은 시간이 소요
- ❖ 정적 시간 분석은 수학적 계산을 통해 모든 경로를 검증하고, 동적 시간 분석은 실제 시뮬레이션을 통해 특정 입력에 대해 성능을 확인하는 방식

4.2. 디지털 회로의 동적 규율

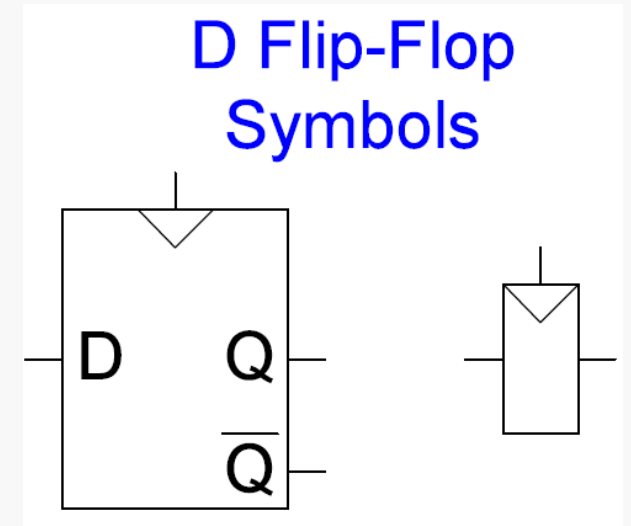
◆ D-플립플롭(Flip/Flop)

❖ 두개의 입력: CLK, D

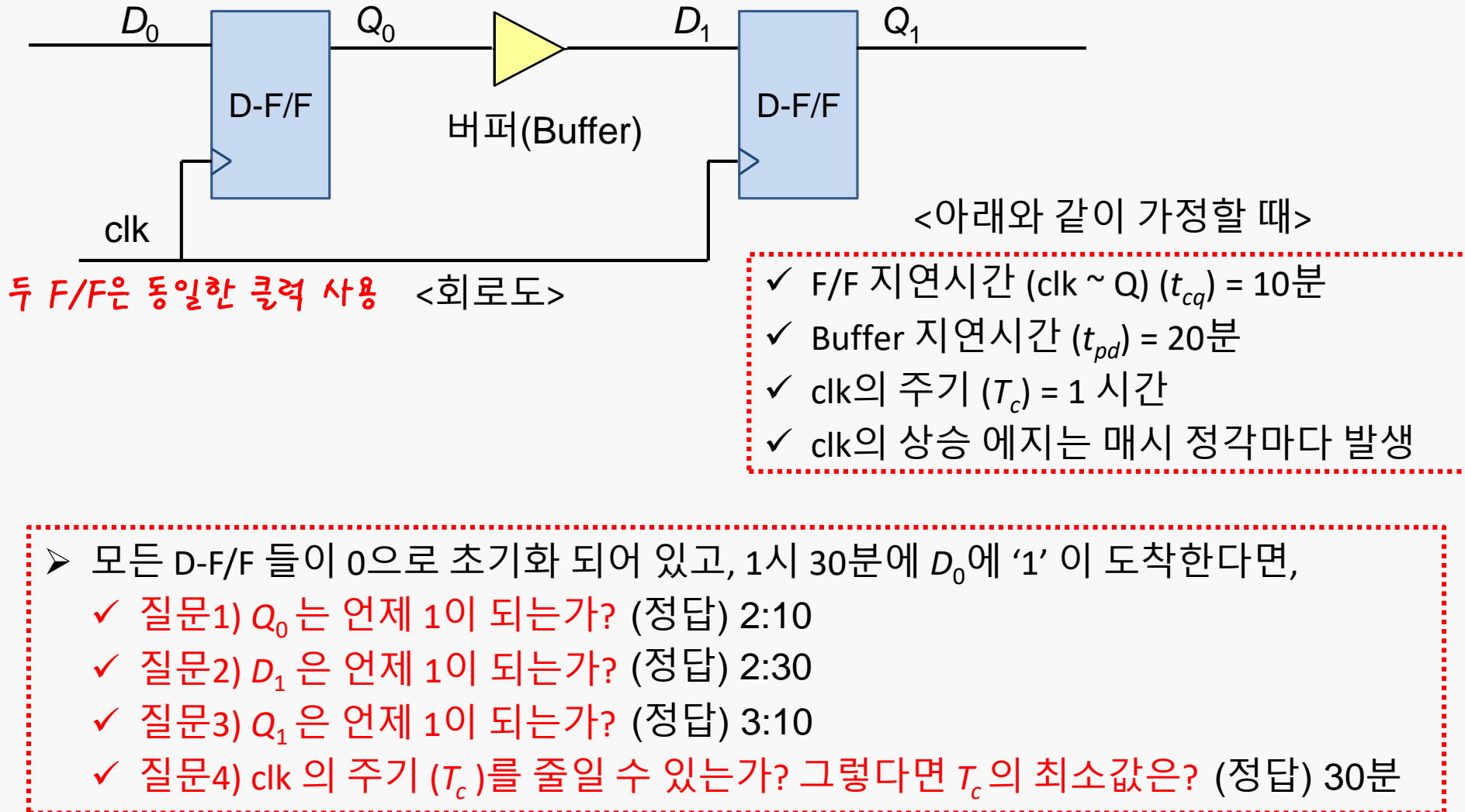
❖ 동작

- 클럭의 상승 에지에서 D 입력에 있는 값이 샘플링 되어 플립플롭 내부에 저장
- 저장된 값은 즉시 Q 출력으로 전달
- 클럭의 에지가 다시 발생하기 전까지 Q 출력은 변하지 않고, 저장된 값을 계속 유지
- 다음 클럭 에지에서 다시 입력 D를 샘플링하고, 이를 출력 Q에 전달

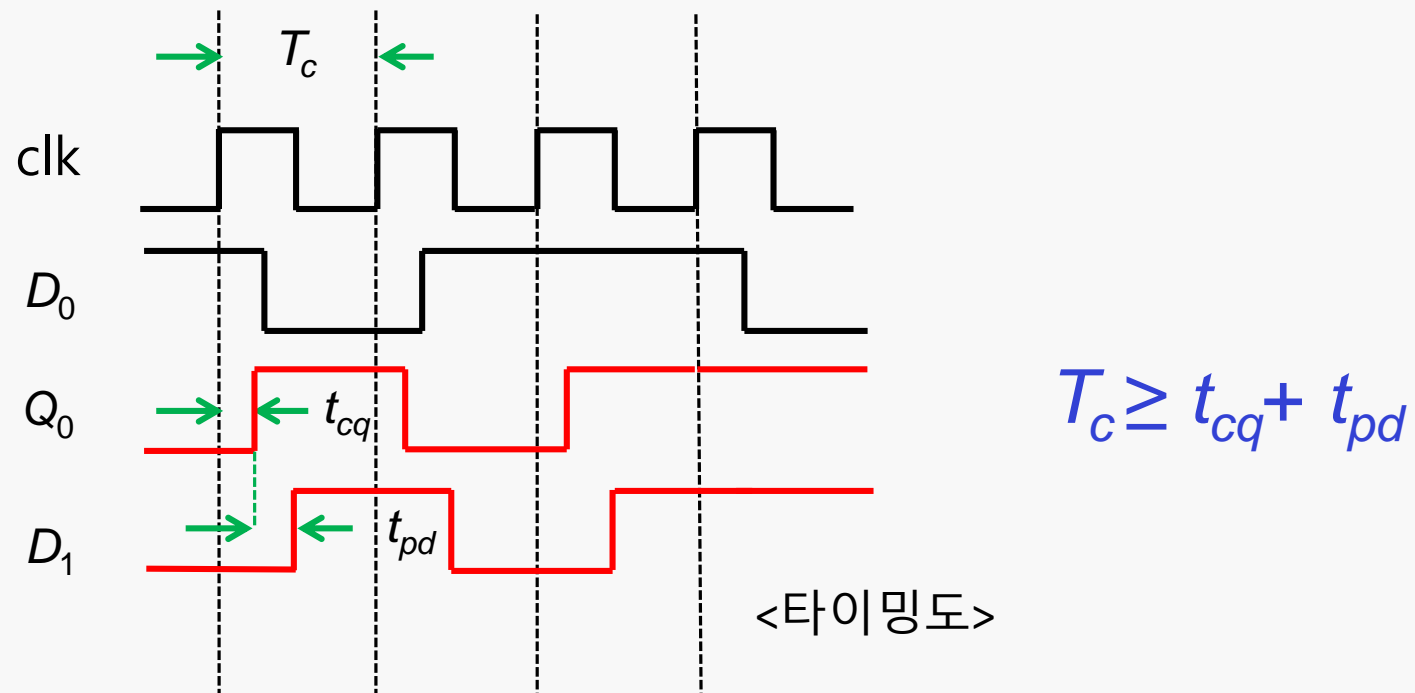
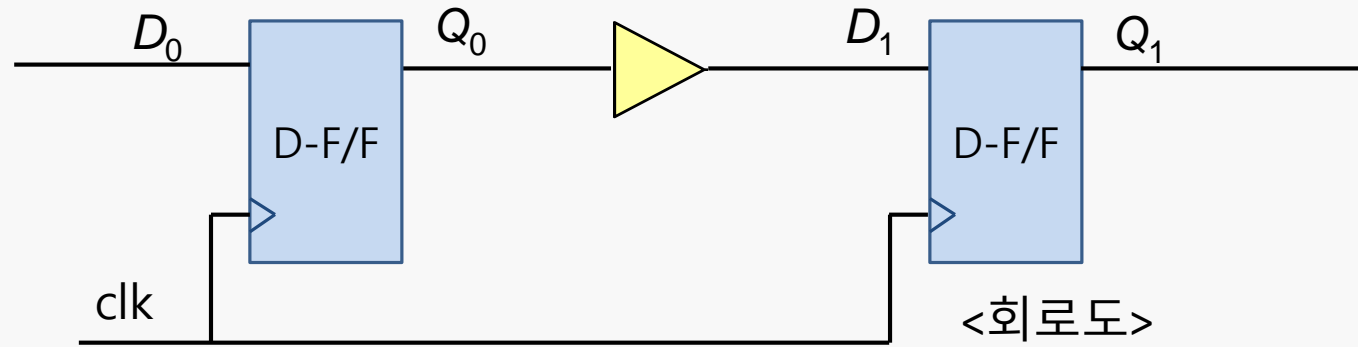
❖ 플립플롭은 항상 클럭 신호의 에지에서 D 입력을 샘플링 하므로, D 입력은 클럭 신호의 에지 발생 전후로 일정한 값을 유지해야 함



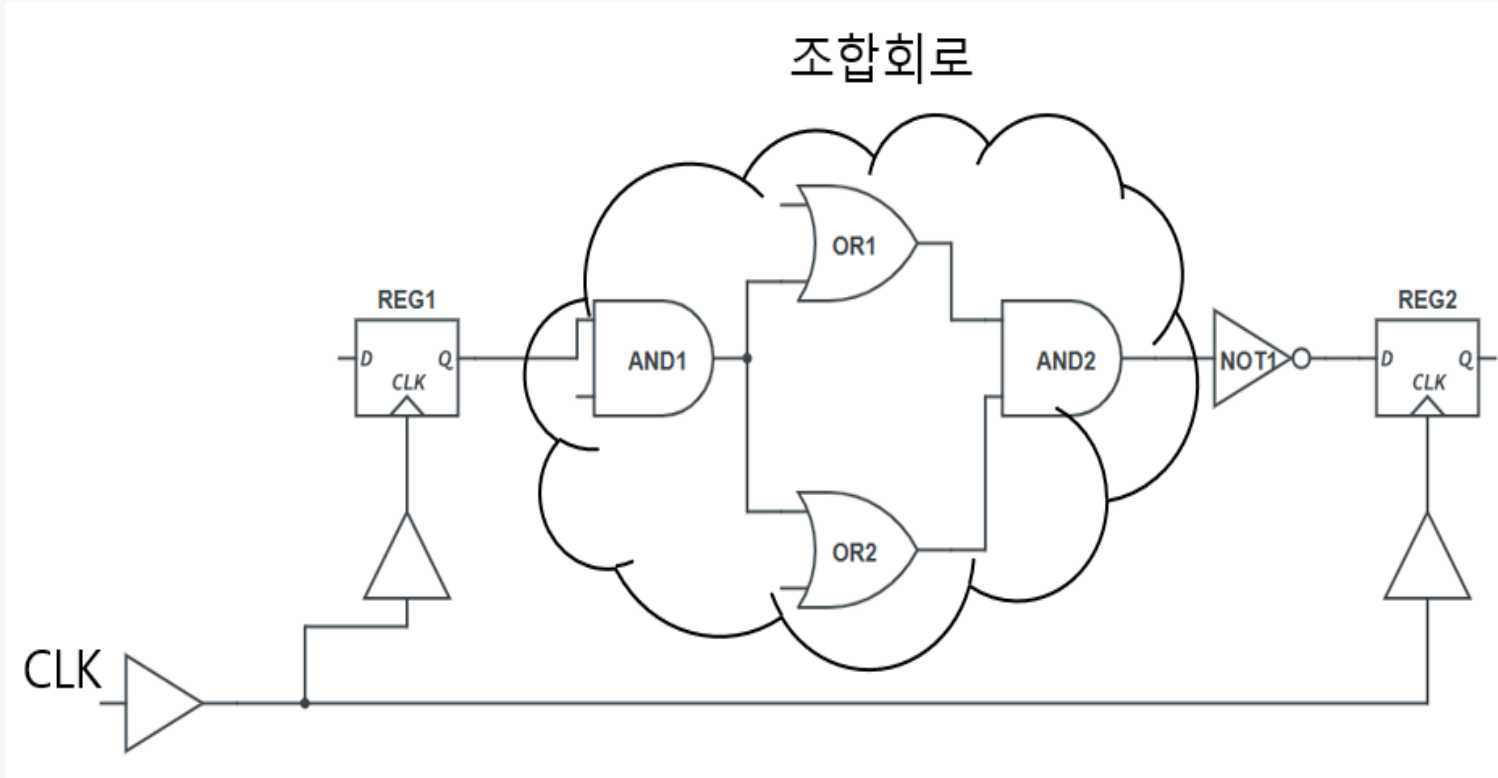
4.2. 디지털 회로의 동적 규율



4.2. 디지털 회로의 동적 규율



4.2. 디지털 회로의 동적 규율



- 클럭의 주기는 Flip-Flop의 지연시간과 조합회로의 지연시간의 합보다 커야 한다.

$$T_c \geq t_{cq} + t_{pd}$$

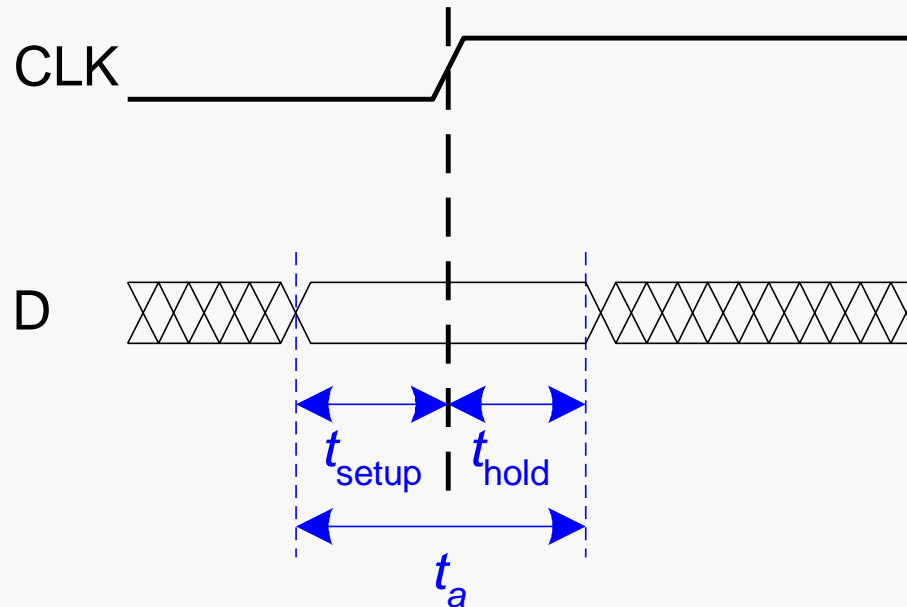
4.2. 디지털 회로의 동적 규율

◆ Setup Time & Hold Time

- ❖ Flip-flop은 항상 클럭의 에지에서 입력 D 를 샘플링 함.
- ❖ 입력 D 는 클럭의 에지 전후로 값이 변하면 안됨.
- ❖ 클럭의 에지에서 입력 D 가 변하면 안됨 = 카메라 셔터를 누르는 순간에 물체가 움직이면 안됨

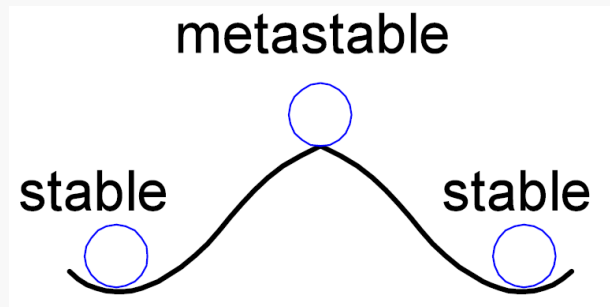
4.2. 디지털 회로의 동적 규율

- ◆ 셋업 타임 (setup time, t_{setup}) 조건: 입력 신호 (D)가 클럭의 에지 발생 전 일정 시간 동안 안정된 상태로 유지되어야 한다는 조건
- ◆ 홀드 타임 (hold time, t_{hold}) 조건이란, 클럭의 에지가 발생한 후에도 일정 시간 동안 입력 신호가 안정된 상태로 유지되어야 한다는 조건



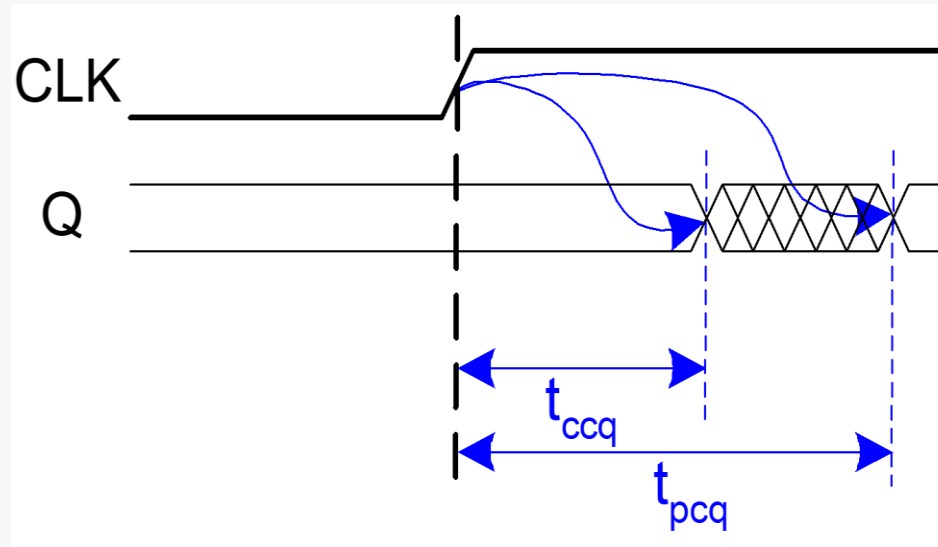
4.2. 디지털 회로의 동적 규율

- ◆ 셋업타임과 홀드타임 조건을 만족하지 못한다면, 플립플롭은 일시적으로 안정된 0 또는 1 상태에 도달하지 못함
- ◆ 이 경우 플립플롭은 불안정한 상태에 머무르다가 결국 안정된 상태로 전환되지만, 이 과정에서 신호의 처리 지연이나 오류가 발생함
- ◆ 메타안정상태 (metastability): 타이밍 위반으로 인해 플립플롭이 일시적으로 빠지게 되는 불안정한 상태
- ◆ 플립플롭의 동적 규율 (dynamic discipline): 플립플롭이 메타안정상태에 빠지는 것을 막기 위해 디지털 회로 내의 모든 플립플롭은 셋업타임 조건과 홀드타임 조건을 준수하도록 설계되어야 함



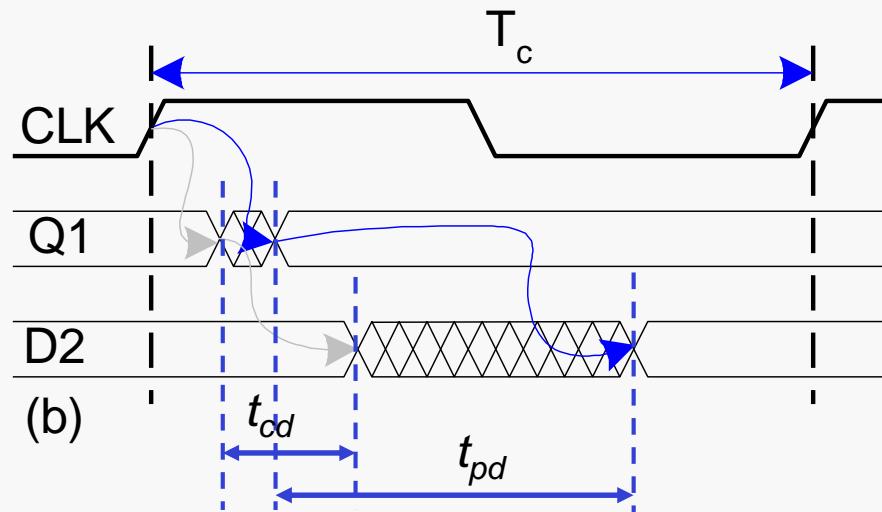
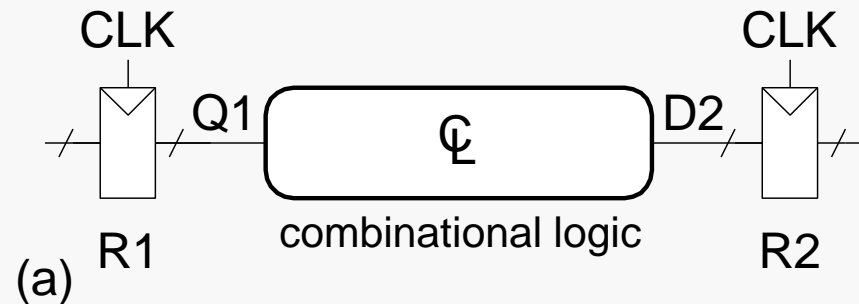
4.2. 디지털 회로의 동적 규율

- ◆ 전파지연 (Propagation Delay, t_{pcq}): 클럭의 에지가 발생한 후, 플립플롭의 입력 (D)이 출력 (Q)으로 전달하는 데 걸리는 **최대 지연시간**
- ◆ 오염지연 (Contamination Delay, t_{ccq}): 클럭의 에지가 발생한 후, 플립플롭의 입력 (D)이 출력 (Q)으로 전달하는 데 걸리는 **최소 지연시간**



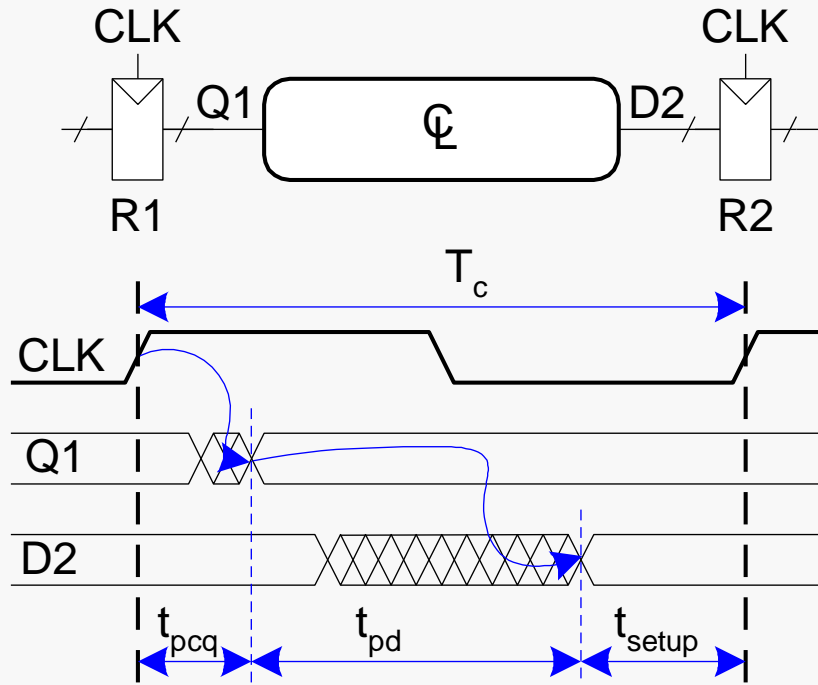
4.2. 디지털 회로의 동적 규율

- ◆ 조합회로의 전파지연 (t_{pd}): 출력이 입력 변화에 맞춰 완전히 바뀌기까지 발생하는 **최대 지연 시간**
- ◆ 조합회로의 오염지연 (t_{cd}): 입력이 변화했을 때 출력이 가장 빠르게 변화하기 시작하는 시점까지 걸리는 **최소 지연 시간**



4.2. 디지털 회로의 동적 규율

- ◆ 셋업타임 조건: 신호가 레지스터 R1에서 출발하여 조합회로를 거쳐 레지스터 R2에 도달하기까지 최대 지연시간과 관련이 있음
- ◆ 레지스터 R2의 입력은 최소 다음 클럭의 에지가 발생하기 셋업 타임 (t_{setup}) 전에 안정화되어야 함



R1의 전파지연 R2의 셋업 타임

what is T_c ?

$$T_c \geq t_{pcq} + t_{pd} + t_{setup}$$

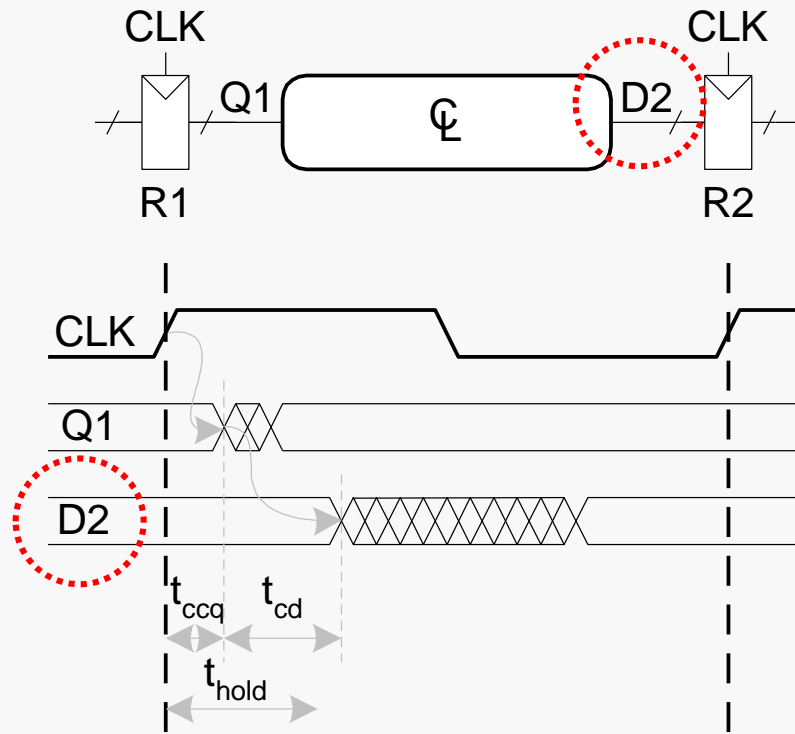
$$t_{pd} \leq T_c - (t_{pcq} + t_{setup})$$

- t_{pcq} , t_{setup} 는 제조사에 의해 정해짐
- T_c 는 설계 조건에 의해서 주어짐
- 주어진 클럭의 주기 안에서 설계자는 조합회로의 전파 지연시간인 t_{pd} 를 줄여 셋업 타임 조건식을 만족하도록 설계해야 함

➤ 셋업타임 조건의 만족을 위해서는 T_c 를 늘리거나, t_{pd} 를 줄여야 함

4.2. 디지털 회로의 동적 규율

- ◆ 홀드타임 조건은 경로의 최소 지연시간과 관련이 있음
- ◆ D2는 클럭의 에지가 발생한 후 $t_{ccq} + t_{cd}$ 이내에는 값이 변하지 않음. R2의 입력은 클럭의 에지가 발생한 후, 적어도 t_{hold} 동안은 일정한 값을 유지해야 함.



R1의 오염지연

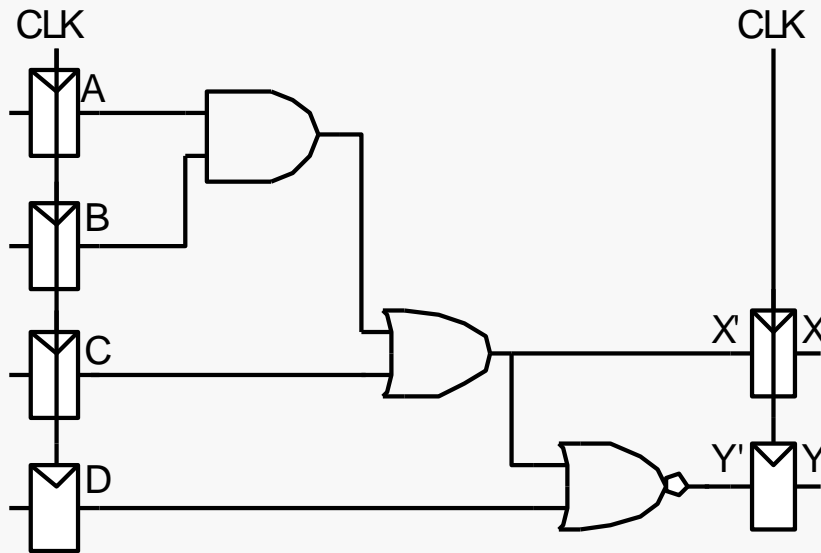
R2의 홀드 타임

$$t_{ccq} + t_{cd} \geq t_{hold}$$
$$t_{cd} \geq t_{hold} - t_{ccq}$$

- t_{hold} , t_{ccq} 는 제조사에 의해 주어짐
- 설계자는 t_{cd} 를 늘려 이 식이 만족되도록 설계해야 함.

➤ 홀드타임 조건의 만족을 위해서는 t_{cd} 를 늘리기 위해 버퍼(buffer)를 삽입해야 함

4.2. 디지털 회로의 동적 규율



$$t_{pd} = 3 \times 40 \text{ ps} = 120 \text{ ps}$$

셋업 타임 조건:

$$T_c \geq (80 + 120 + 50) \text{ ps} = 250 \text{ ps}$$

$$f_c = 1/T_c \leq 4.0 \text{ GHz}$$

주어진 타이밍 조건

$$\text{Per F/F} \left[\begin{array}{l} t_{ccq} = 30 \text{ ps} \\ t_{pcq} = 80 \text{ ps} \\ t_{setup} = 50 \text{ ps} \\ t_{hold} = 60 \text{ ps} \end{array} \right.$$

$$\text{per gate} \left[\begin{array}{l} t_{pd} = 40 \text{ ps} \\ t_{cd} = 25 \text{ ps} \end{array} \right.$$

$$t_{cd} = 25 \text{ ps}$$

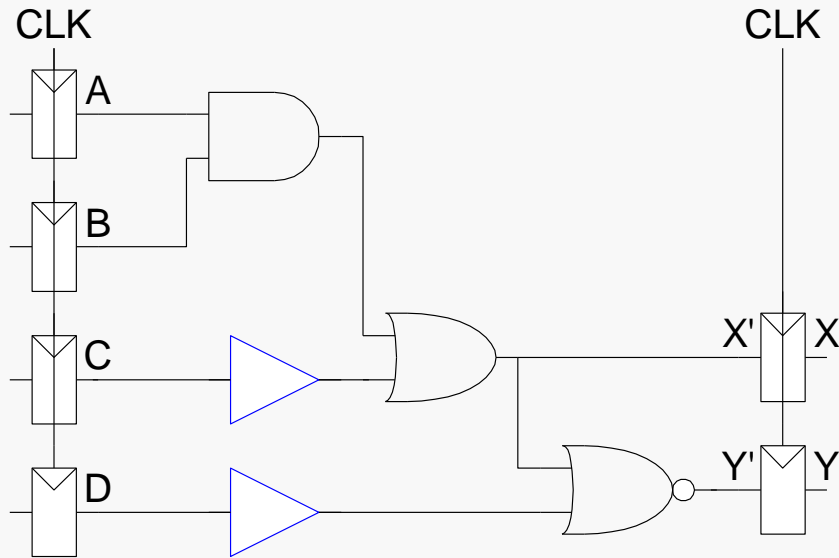
홀드 타임 조건:

$$t_{ccq} + t_{cd} \geq t_{hold} ?$$

$(30 + 25) \text{ ps} \geq 60 \text{ ps} ?$ **홀드타임이 충분치 않음!**

4.2. 디지털 회로의 동적 규율

짧은 경로에 버퍼 추가



$$t_{pd} = 3 \times 40 \text{ ps} = 120 \text{ ps}$$

셋업타임 조건:

$$T_c \geq (80 + 120 + 50) \text{ ps} = 250 \text{ ps}$$

$$f_c = 1/T_c \leq 4.0 \text{ GHz}$$

주어진 타이밍 조건

$$\begin{array}{l} \text{Per F/F} \\ \left[\begin{array}{l} t_{ccq} = 30 \text{ ps} \\ t_{pcq} = 80 \text{ ps} \\ t_{setup} = 50 \text{ ps} \\ t_{hold} = 60 \text{ ps} \end{array} \right. \end{array}$$

$$\begin{array}{l} \text{per gate} \\ \left[\begin{array}{l} t_{pd} = 40 \text{ ps} \\ t_{cd} = 25 \text{ ps} \end{array} \right. \end{array}$$

$$t_{cd} = 2 \times 25 \text{ ps} = 50 \text{ ps}$$

홀드타임 조건:

$$t_{ccq} + t_{cd} \geq t_{hold} ?$$

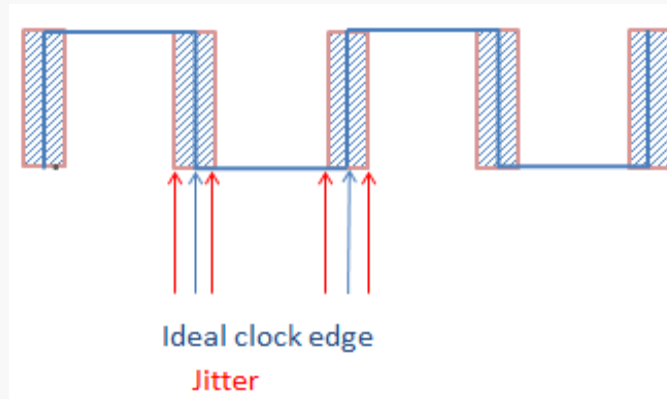
$$(30 + 50) \text{ ps} \geq 60 \text{ ps} ? \text{ 홀드타임이 충분!}$$

✓ 버퍼는 back end 과정에서 삽입함

4.2. 디지털 회로의 동적 규율

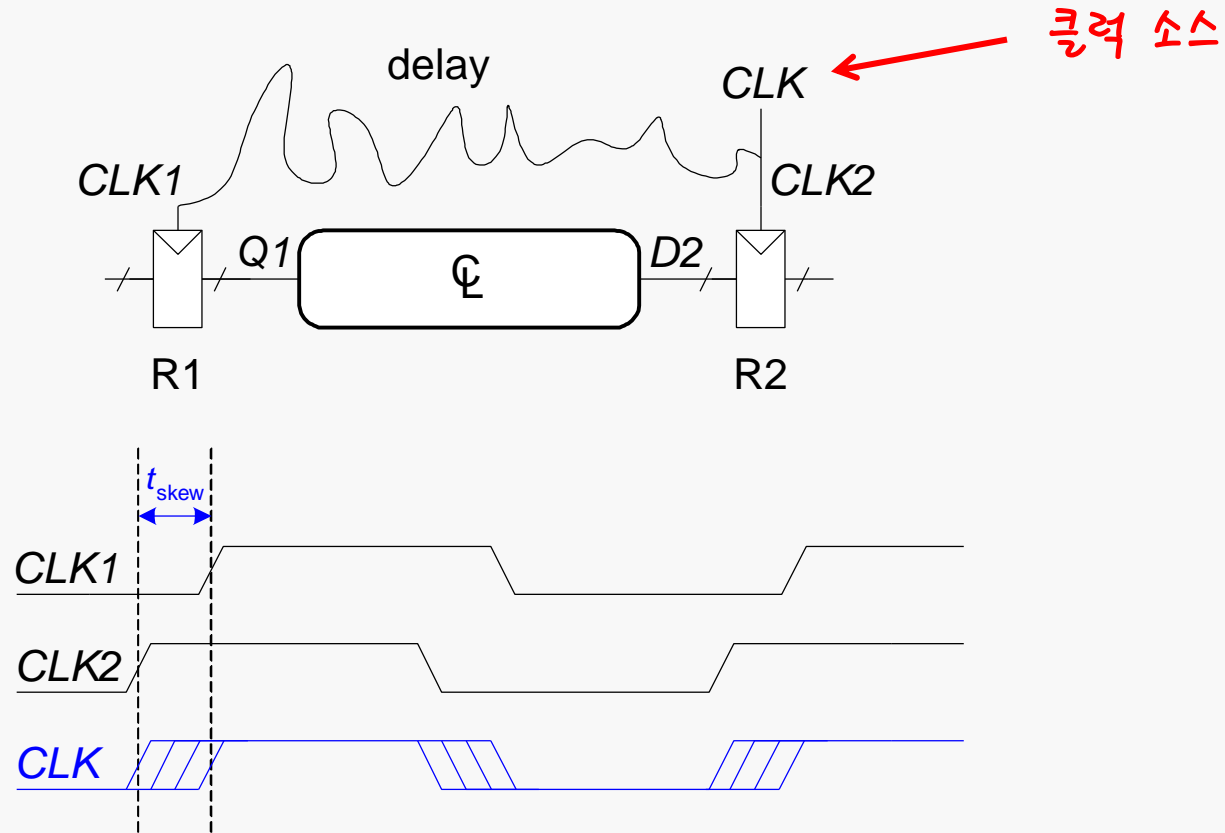
- ◆ 클럭의 불확실성 (clock uncertainty): 클럭 신호가 이상적인 시점에 도착하지 않고 예상보다 빠르거나 느리게 도착할 수 있는 시간 변동을 의미
- ◆ 클럭 스큐 (skew), 지터 (jitter), 타이밍 마진 (margin)
 - ❖ Jitter: 이상적인 클럭 주기에서의 시간적 변동으로, 신호의 주파수 안정성과 타이밍을 저하시키는 원인이 됨
 - ❖ Margin: skew와 jitter로 인한 타이밍 변화를 보완하기 위해 설정된 여유 시간으로, 안정적인 동작을 보장하기 위한 설계 여유를 의미함
- ◆ 각각의 요소는 디지털 회로의 타이밍에 중요한 영향을 미치며, 회로가 정확하게 동작하도록 보장하기 위해 설계시 반드시 고려되어야 함

클럭 지터

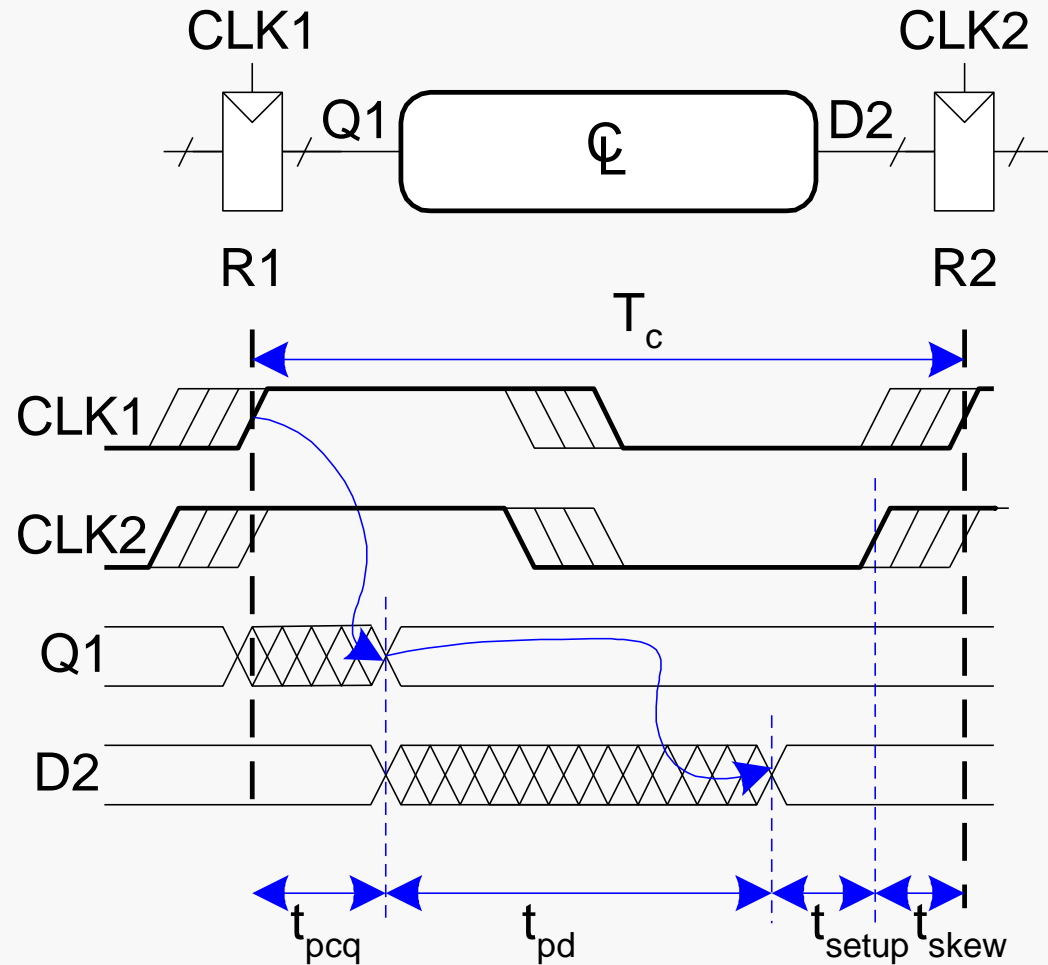


4.2. 디지털 회로의 동적 규율

- ◆ 클럭 스큐 (skew): 동일한 클럭 신호가 서로 다른 위치에 있는 플립플롭들에 전달될 때 발생하는 시간 차이



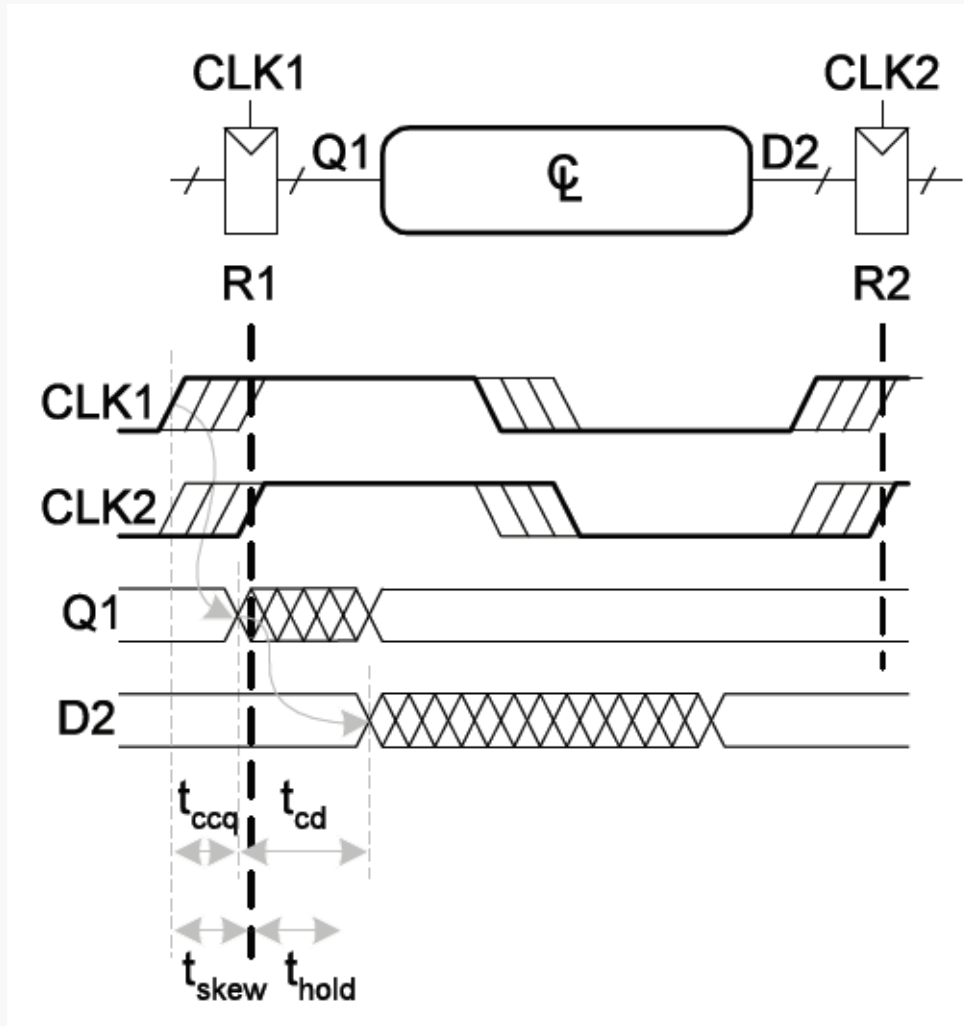
4.2. 디지털 회로의 동적 규율



CLK2가 CLK1보다 빠르게 도착

$$T_c \geq t_{pcq} + t_{pd} + t_{setup} + t_{skew}$$
$$t_{pd} \leq T_c - (t_{pcq} + t_{setup} + t_{skew})$$

4.2. 디지털 회로의 동적 규율

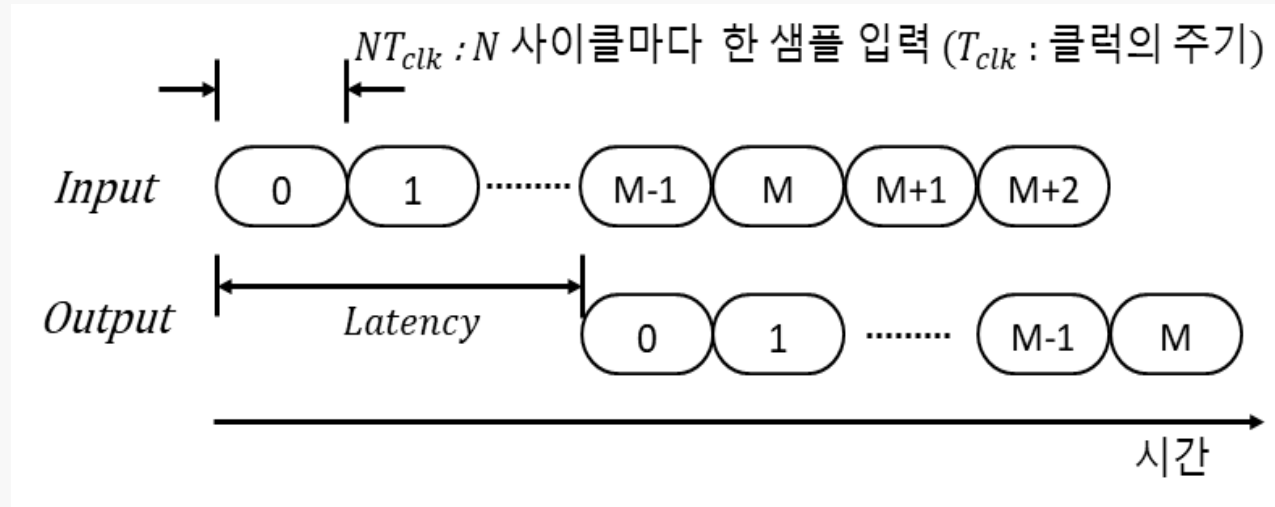


CLK2가 CLK1보다 느리게 도착

$$t_{ccq} + t_{cd} \geq t_{hold} + t_{skew}$$

$$t_{cd} \geq t_{hold} + t_{skew} - t_{ccq}$$

4.3. 디지털 회로의 가속 설계 방법



◆ 처리량 (Throughput)

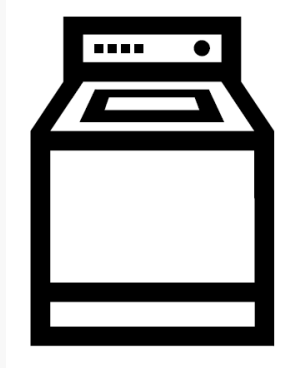
- ❖ 시스템이 초당 얼마나 많은 양을 처리할 수 있는지를 나타냄
- ❖ $1/T_p = 1/(NT_{clk})$ [samples /sec]

◆ 레이턴시 (Latency)

- ❖ 하나의 작업이 처리되는 데 걸리는 시간을 의미
- ❖ $MT_p = MNT_{clk}$ [sec]

4.3. 디지털 회로의 가속 설계 방법

입력



- ◆ 세탁기
- ◆ 세탁기의 지연 시간 ($Washer_{PD}$) = 30분

출력



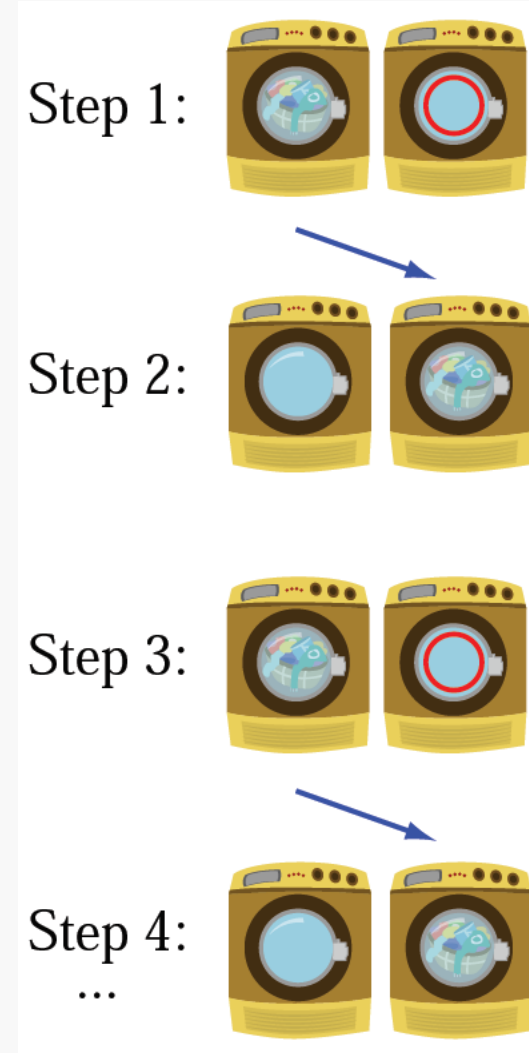
- ◆ 건조기
- ◆ 건조기의 지연 시간 ($Dryer_{PD}$) = 60분

4.3. 디지털 회로의 가속 설계 방법

◆파이프라이닝(Pipelining)을 적용하지 않을 때



$$\begin{aligned}\text{총 처리시간} &= N * (\text{Washer}_{PD} + \text{Dryer}_{PD}) \\ &= \underline{N * 90} \text{ mins}\end{aligned}$$

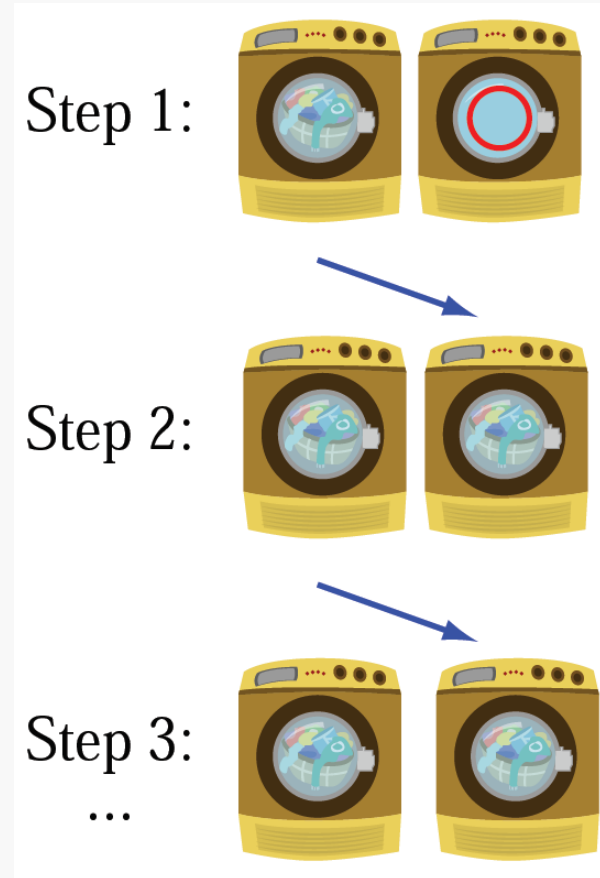


4.3. 디지털 회로의 가속 설계 방법

◆파이프라이닝 (Pipelining)을 적용할 때

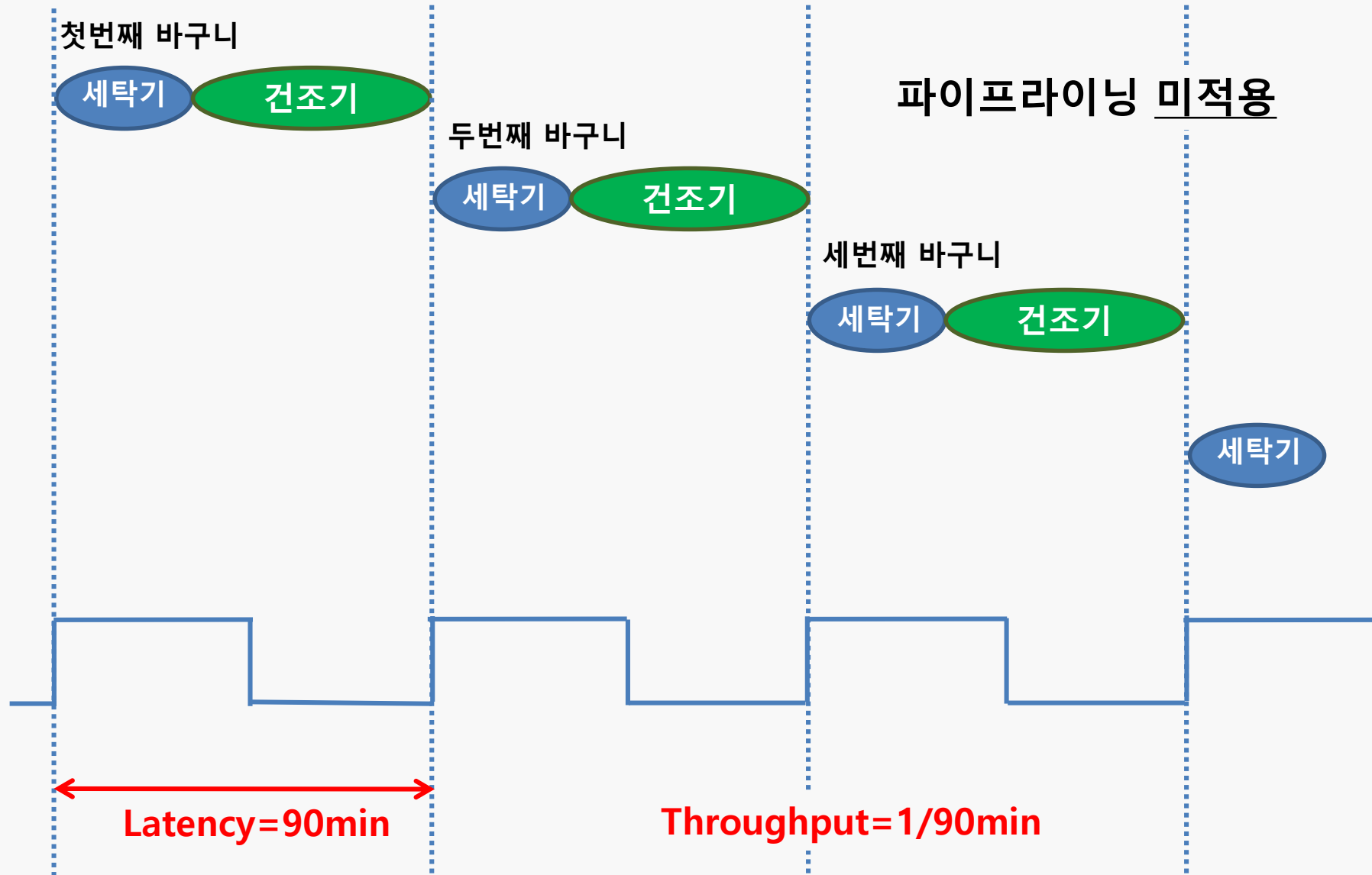


$$\begin{aligned}\text{총 처리시간} &= N * \text{Max}(\text{Washer}_{PD}, \text{Dryer}_{PD}) \\ &= \underline{N*60 + 30} \text{ mins}\end{aligned}$$

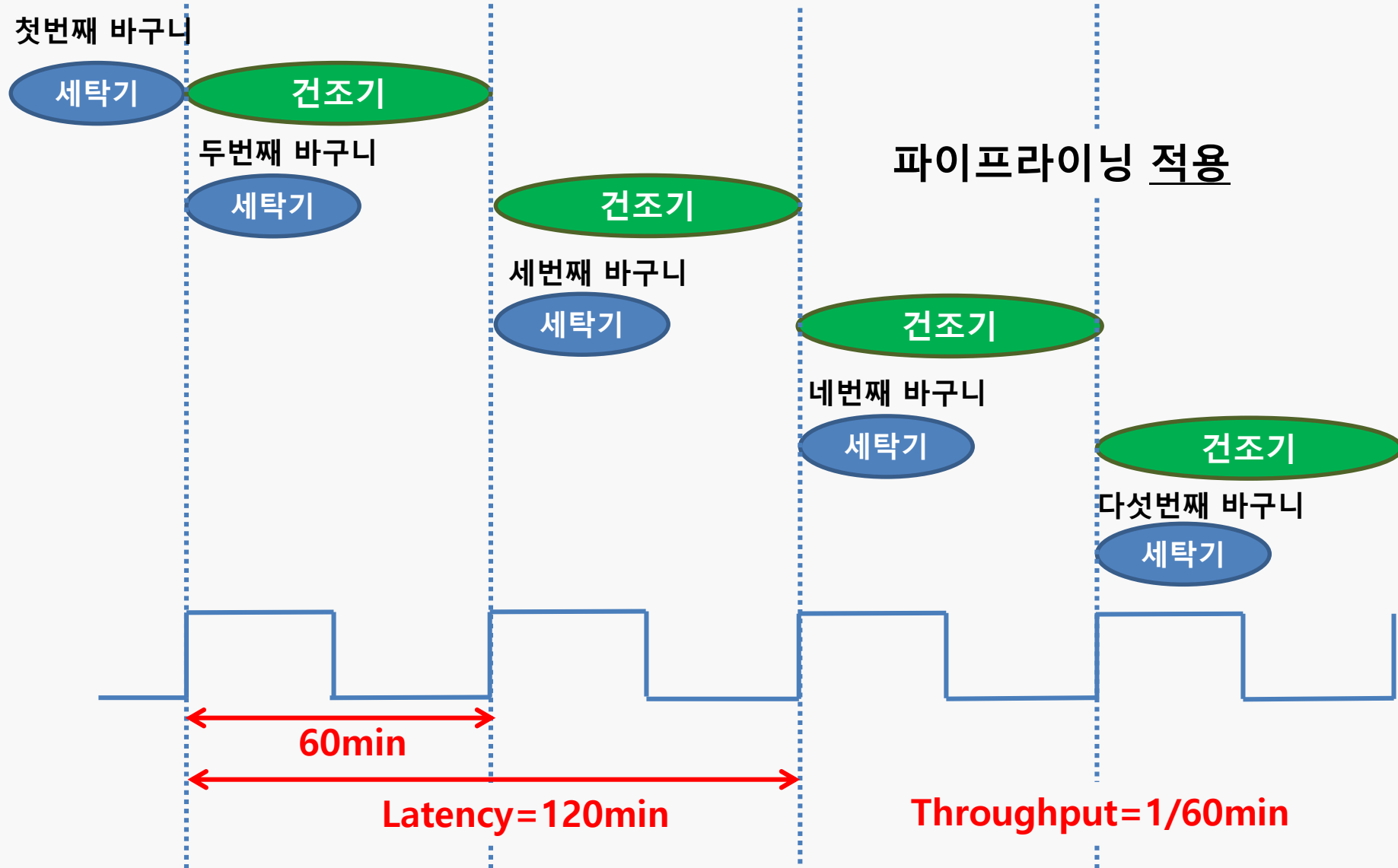


****입력 바구니의 개수가 아주 많다고 가정하면 총 지연시간은 N*60 mins 에 수렴함**

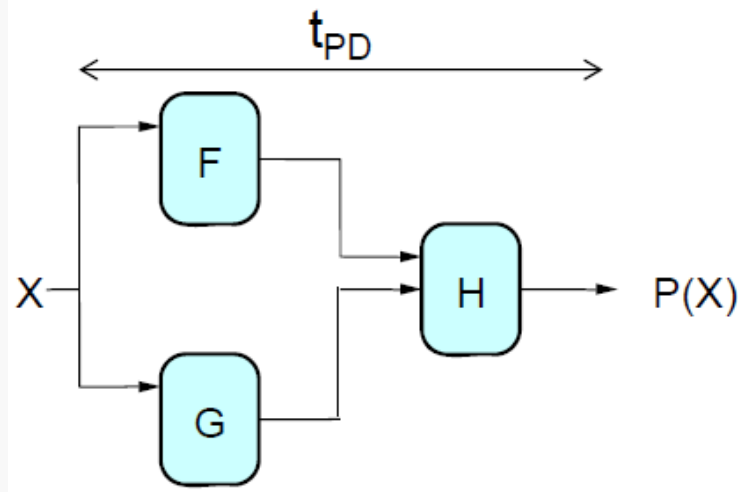
4.3. 디지털 회로의 가속 설계 방법



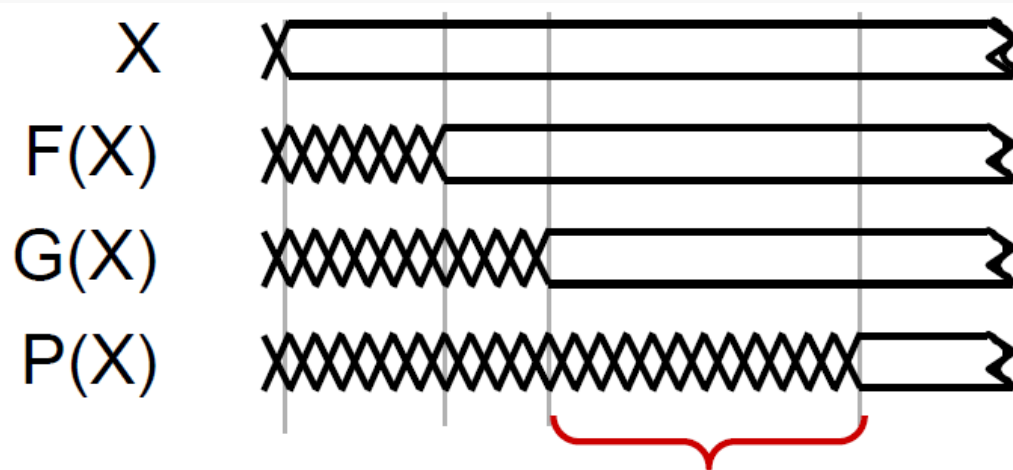
4.3. 디지털 회로의 가속 설계 방법



4.3. 디지털 회로의 가속 설계 방법



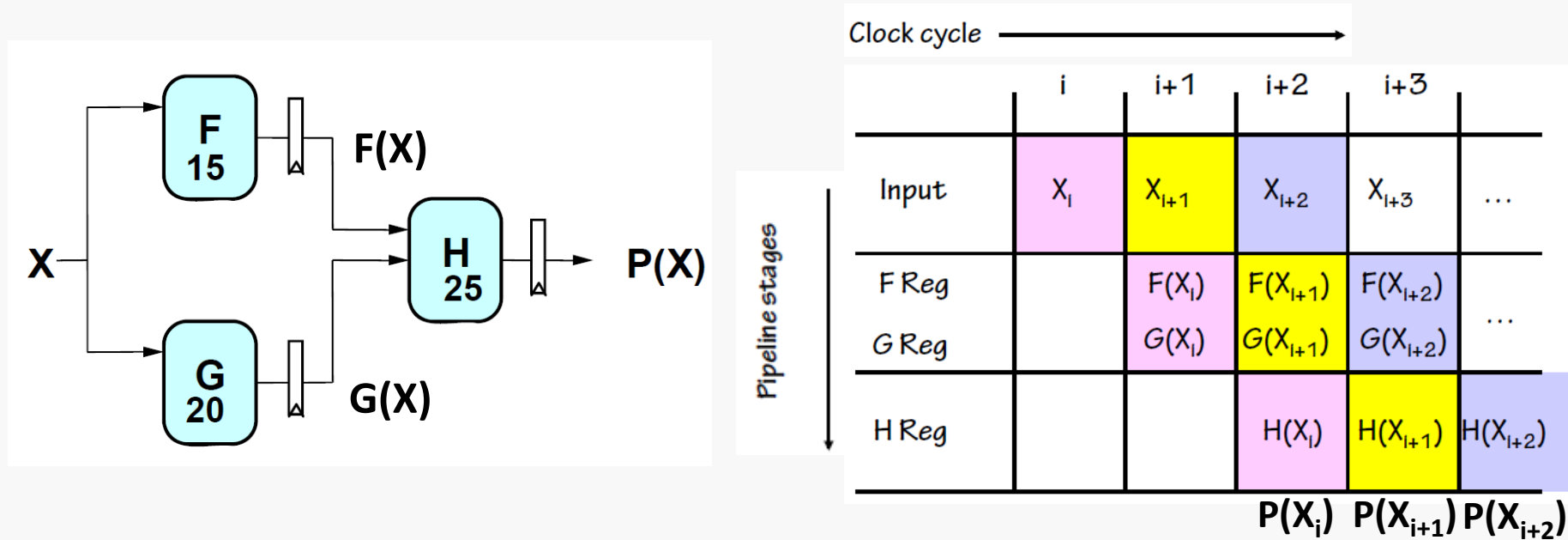
- ❖ Latency: t_{PD}
- ❖ Throughput: $1/t_{PD}$



✓ F와 G의 출력은 H의 입력임

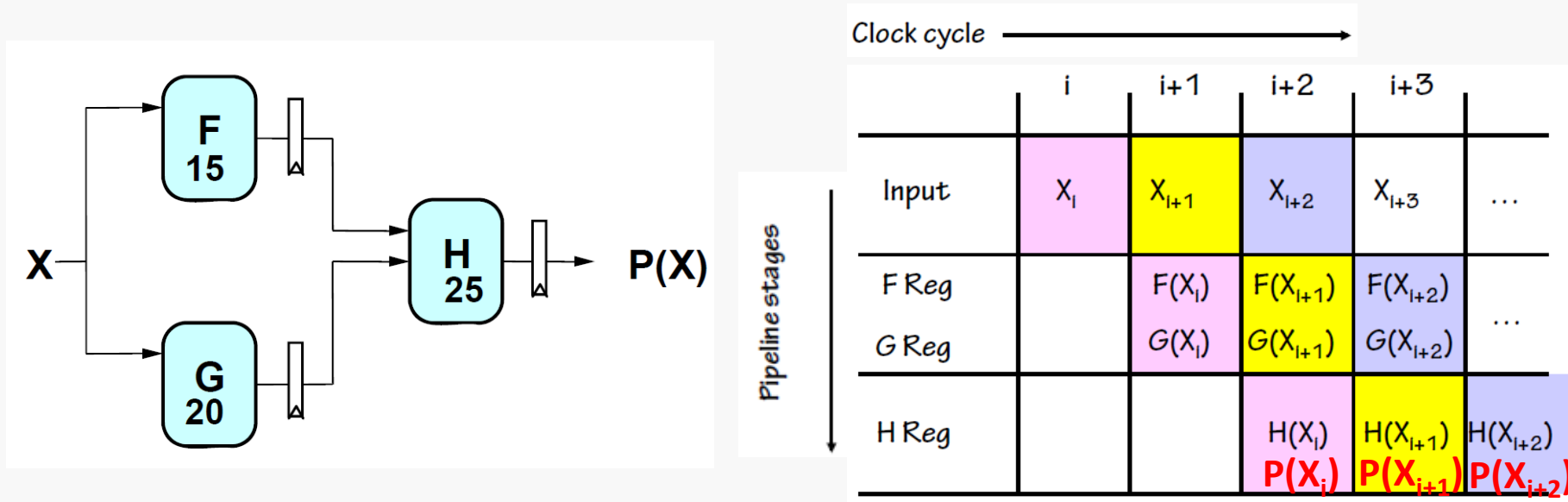
✓ F와 G는 H가 계산을 마칠 때까지 새로운 입력을 받지 말고 idle 상태를 유지하며 출력 값을 계속 유지하고 있어야 함.

4.3. 디지털 회로의 가속 설계 방법



- ❖ 위의 다이어그램에서 특정 입력 데이터 세트에 대한 결과는 대각선 방향으로 이동하며, 각 클럭 사이클마다 하나의 파이프라인 단계를 진행함

4.3. 디지털 회로의 가속 설계 방법



- ❖ F, G, H 함수의 전파지연은 각각 15 ns, 20 ns, 25 ns
- ❖ 레지스터의 전파지연 및 셋업타임 등은 0인 이상적인 레지스터를 가정할 때,

	<u>latency</u>	<u>throughput</u>
unpipelined		
2-stage pipeline	_____	_____
	worse	better

4.3. 디지털 회로의 가속 설계 방법

◆ 파이프라인 단계 (Pipeline stages)

- ❖ ***K-Stage Pipeline*** (“K-pipeline”): 입력에서 출력으로 가는 모든 경로에 정확히 **k 개의 레지스터**를 포함.

◆ Pipeline registers

- ❖ 모든 파이프라인 단계는 **출력에 레지스터**를 가지며 입력에는 없음

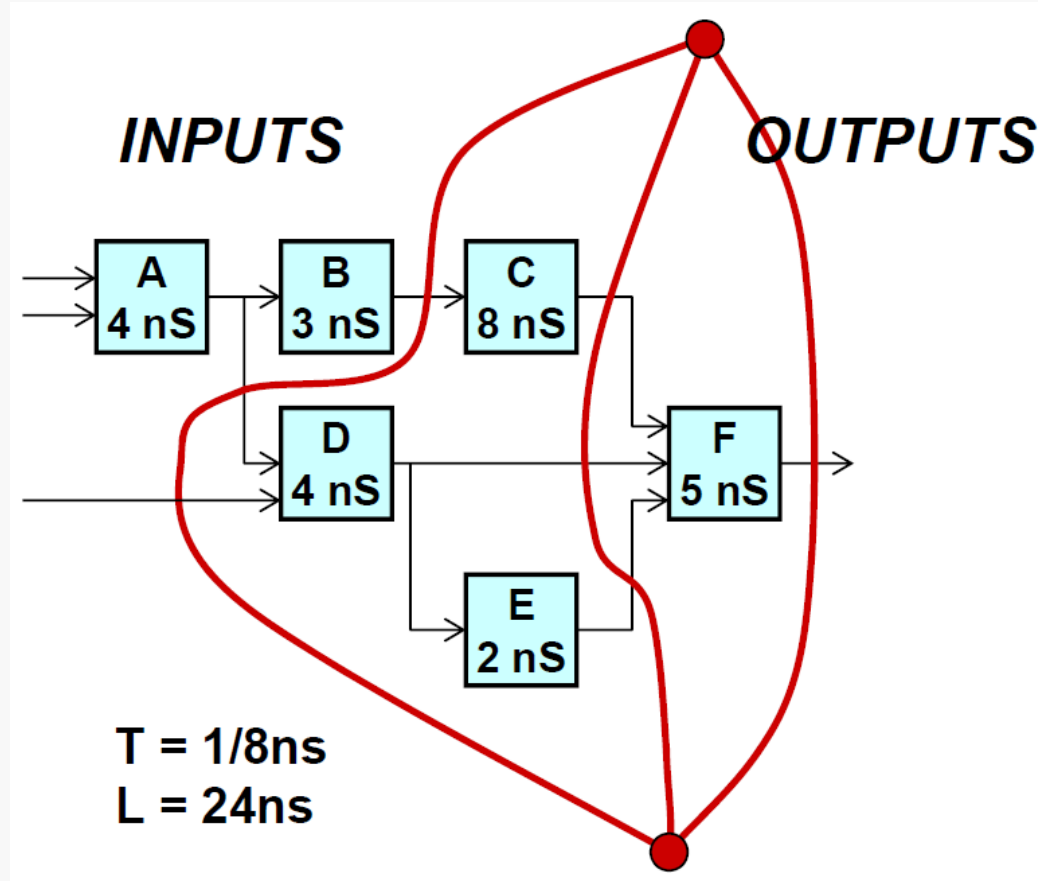
◆ Latency

- ❖ ***K-Stage Pipeline*** 회로의 레이턴시(latency)는 클럭 주기의 **k 배**

◆ Throughput

- ❖ ***K-Stage Pipeline***의 처리량(throughput)은 클럭의 주파수와 동일

4.3. 디지털 회로의 가속 설계 방법



4.3. 디지털 회로의 가속 설계 방법

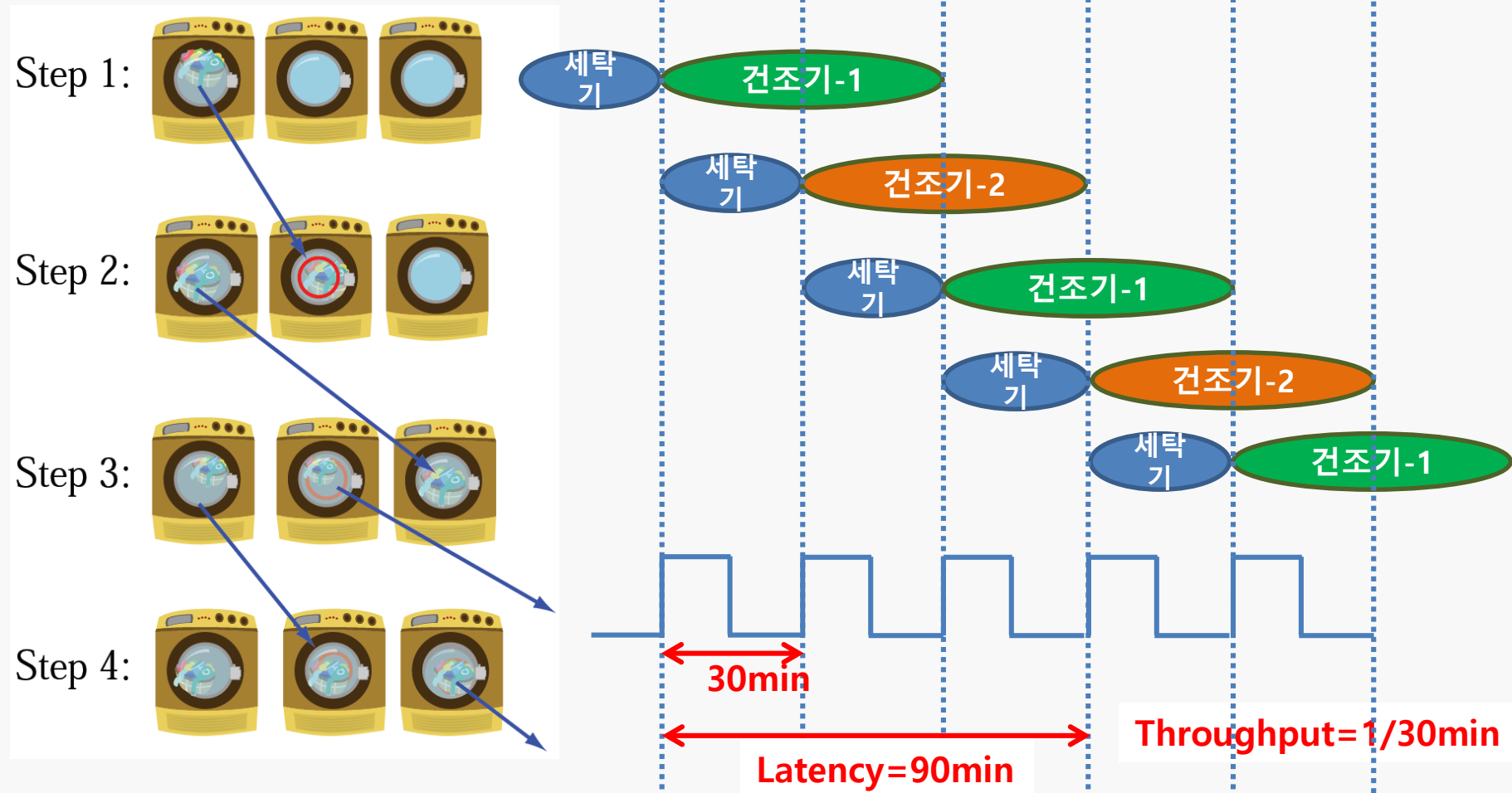
◆ 장점:

- ❖ 임계 경로 (Critical path)의 지연 시간을 줄일 수 있으므로, 더 빠른 클럭 사용 가능 → 처리량 (throughput) 증가

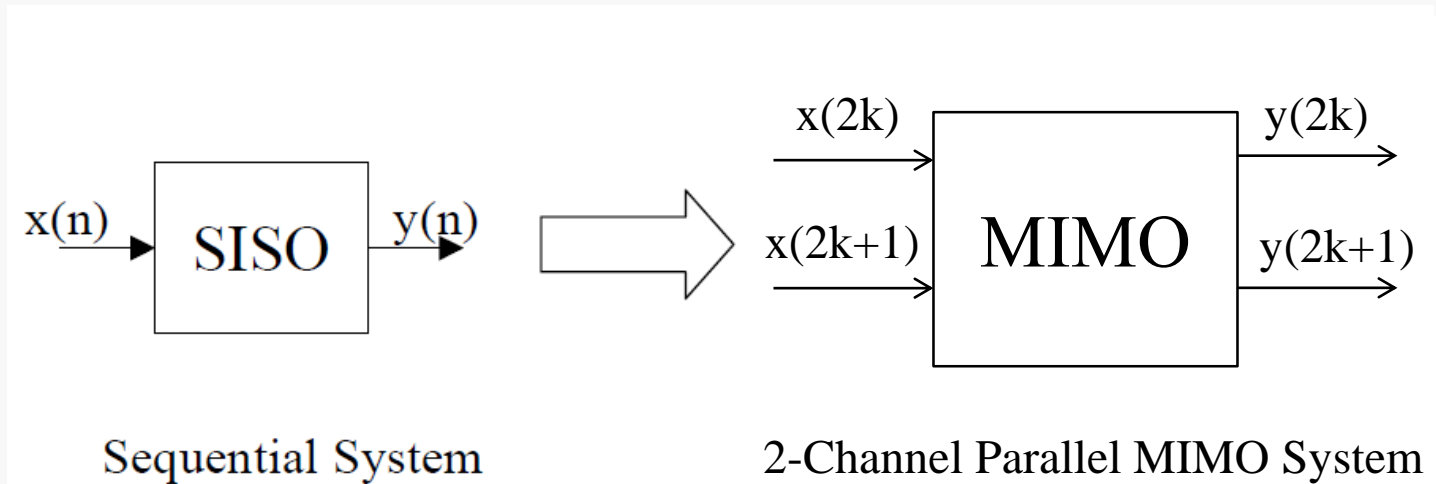
◆ 단점:

- ❖ 파이프라인 레지스터 추가로 인한 면적 증가
- ❖ 레이턴시 (latency) 증가
- ❖ 전력 소모 증가

4.3. 디지털 회로의 가속 설계 방법



4.4. 유한 임펄스 응답 필터의 가속 설계 방법



◆ SISO (single-input single-output) 유한 임펄스 응답 필터 (Finite Impulse Response Filter)

❖ $y(n) = ax(n) + bx(n-1) + cx(n-2)$

◆ MIMO (multiple-input multiple-out) 병렬 처리를 위한 FIR 필터

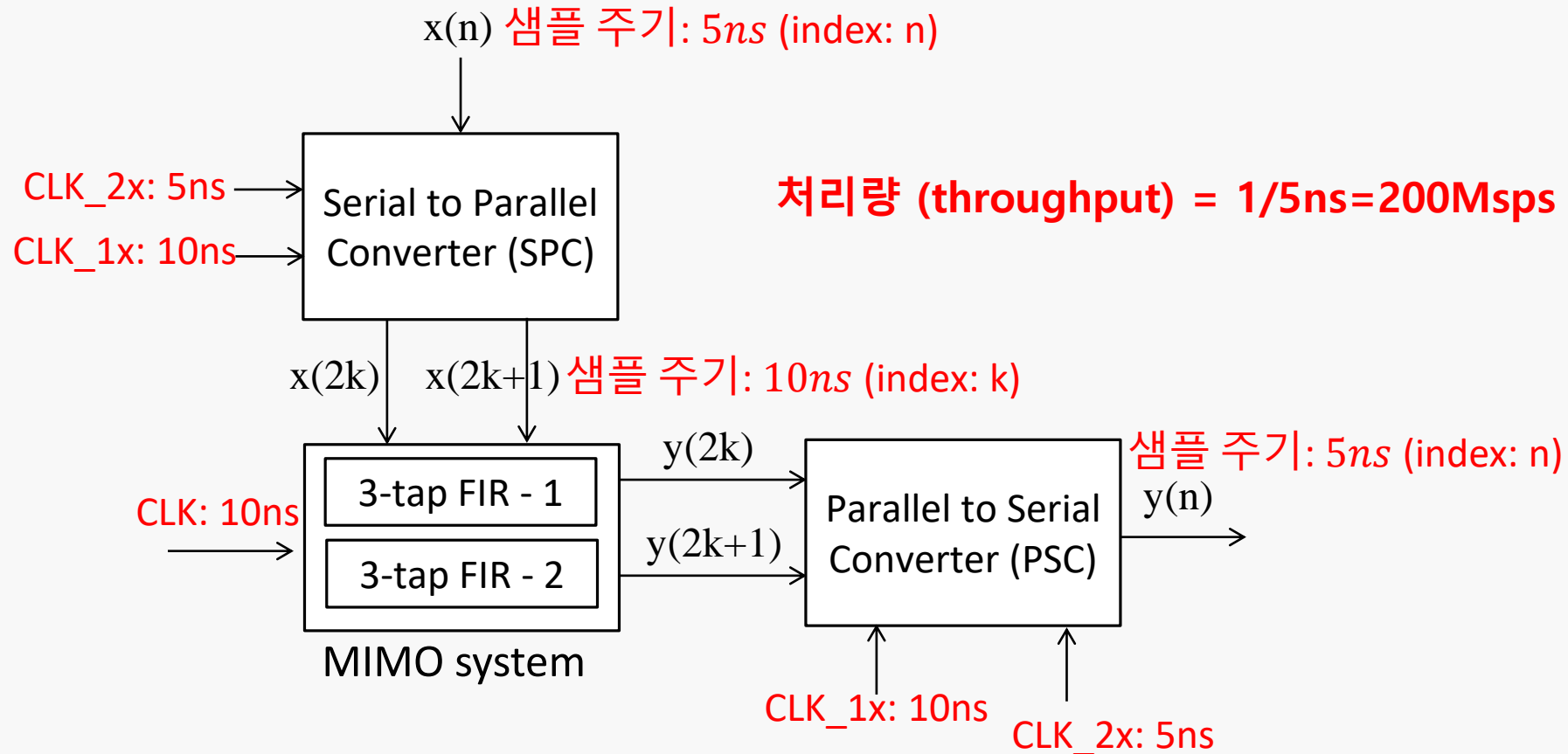
❖ $y(2k) = ax(2k) + bx(2k-1) + cx(2k-2)$

❖ $y(2k+1) = ax(2k+1) + bx(2k) + cx(2k-1)$

4.4. 유한 임펄스 응답 필터의 가속 설계 방법

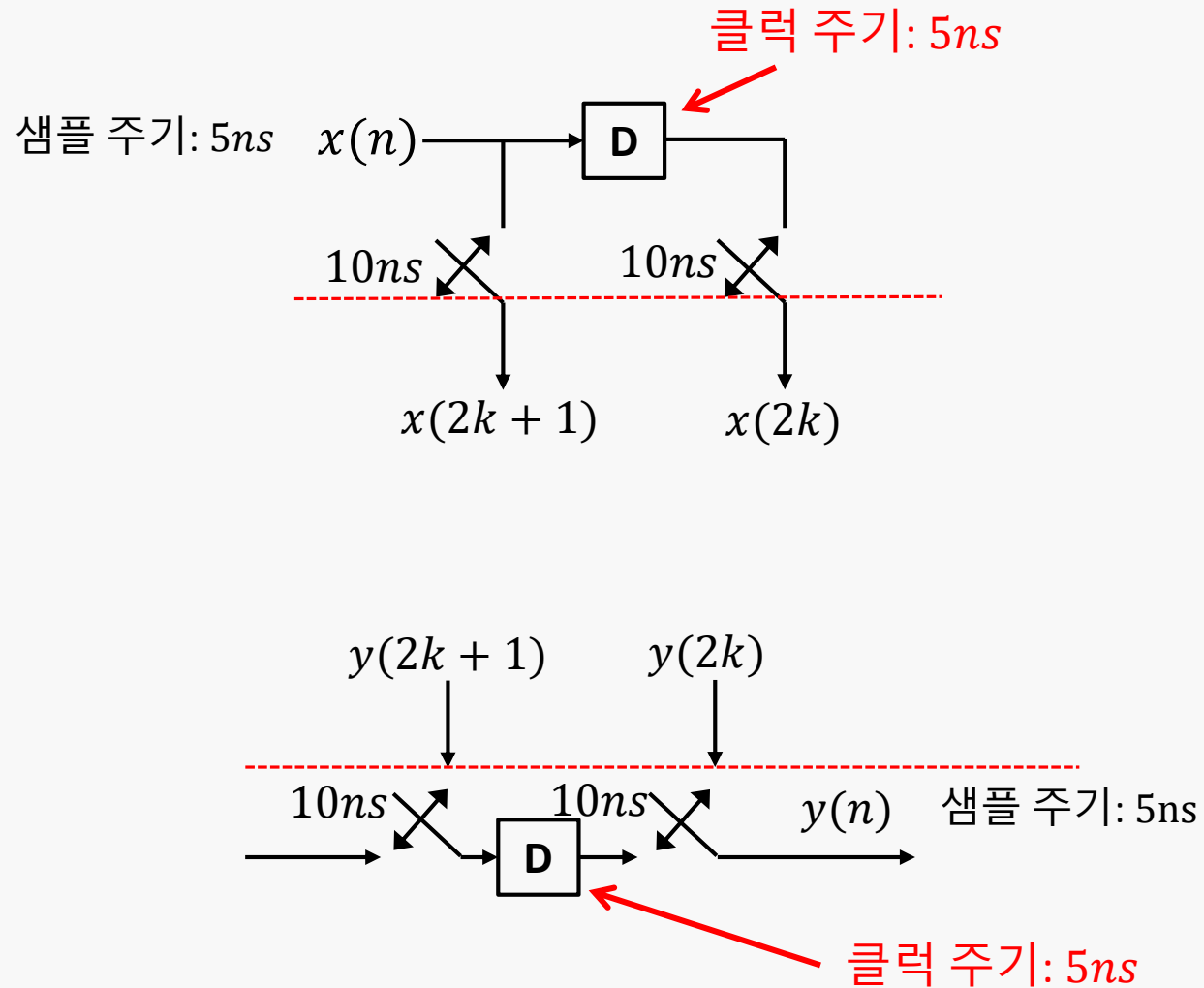
- ◆ k^{th} 클럭 사이클에서,
 - ❖ 2개의 입력 $x(2k)$ 와 $x(2k + 1)$ 이 처리됨.
 - ❖ 2개의 샘플 $y(2k)$ 와 $y(2k + 1)$ 이 생성됨
- ◆ $k = 0^{th}$ 클럭 사이클 : $y(0), y(1)$
- ◆ $k = 1^{st}$ 클럭 사이클 : $y(2), y(3)$ ❖ 면적 $\geq 2x$
- ◆ $k = 2^{nd}$ 클럭 사이클 : $y(4), y(5)$ ❖ 처리량 (throughput) $\leq 2x$

4.4. 유한 임펄스 응답 필터의 가속 설계 방법

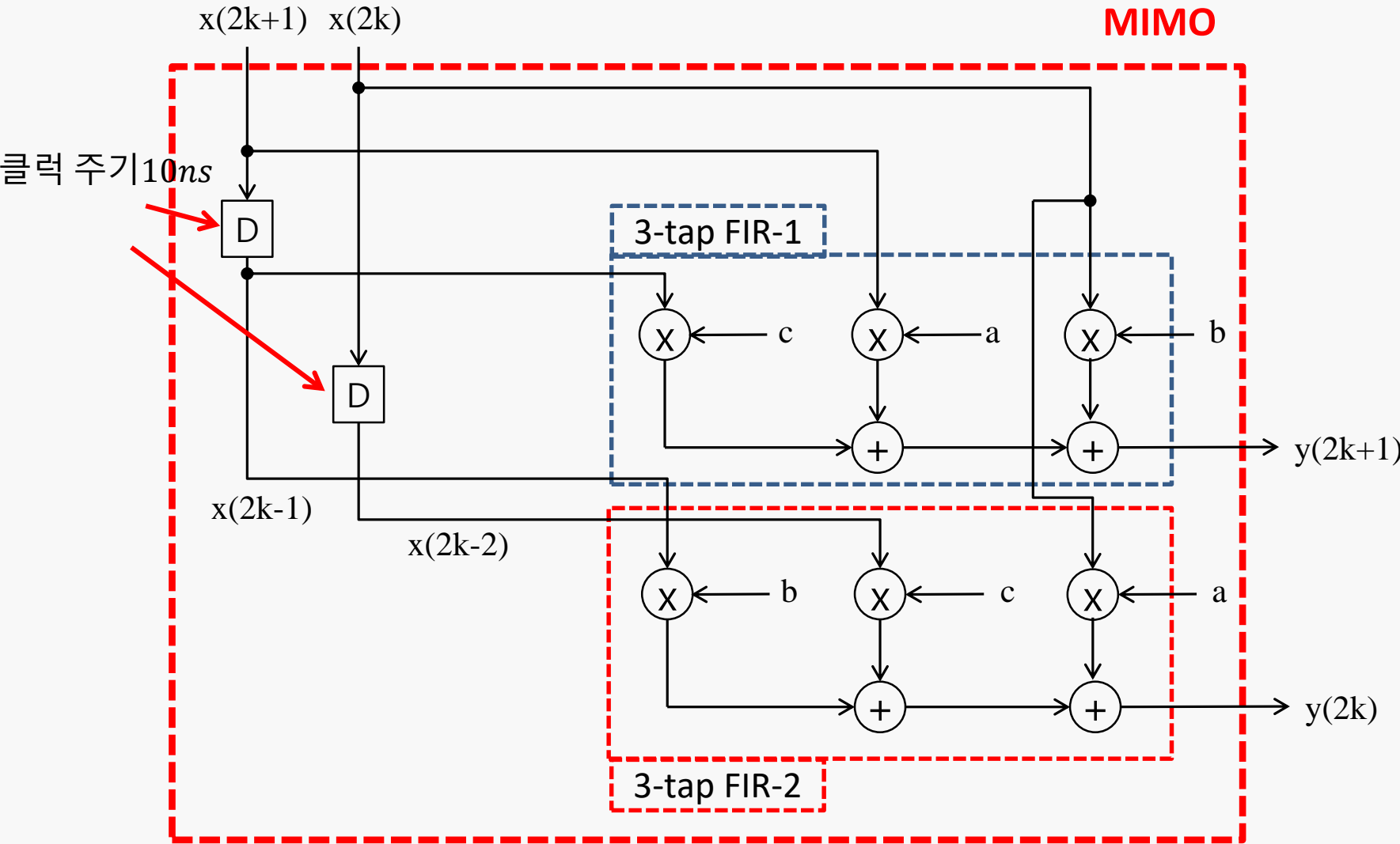


- ✓ SPC는 $5ns$ 마다 입력 $x(n)$ 을 1개씩 받아서, $10ns$ 마다 출력을 2개씩 배출 ($x(2k)$, $x(2k+1)$)
- ✓ 2개의 FIR 필터는 $10ns$ 클럭을 사용하며, $10ns$ 마다 출력을 2개씩 배출 ($y(2k)$, $y(2k+1)$)
- ✓ PSC는 $10ns$ 마다 입력을 2개씩 받아서 $5ns$ 마다 출력 $y(n)$ 을 1개씩 배출

4.4. 유한 임펄스 응답 필터의 가속 설계 방법



4.4. 유한 임펄스 응답 필터의 가속 설계 방법



4.4. 유한 임펄스 응답 필터의 가속 설계 방법

```
module spc(clk_1x, clk_2x, rst, x, y1, y2);
    parameter N = 8;
    input clk_1x, clk_2x, rst;
    input [N-1:0] x;
    output reg [N-1:0] y1, y2;

    reg [2*N-1:0] q;

    always @(posedge clk_2x) begin //fast clk
        if (rst) q <= 0;
        else q <= {q[N-1:0], x};
    end

    always @(posedge clk_1x) begin //slow clk
        if (rst) begin
            y1 <= 0;
            y2 <= 0; end
        else begin
            y1 <= q[N-1:0];
            y2 <= q[2*N-1:N];end
    end
endmodule
```

<Serial-to-Parallel Converter의 Verilog-HDL 기술>

4.4. 유한 임펄스 응답 필터의 가속 설계 방법

<Parallel-to-Serial Converter의
Verilog-HDL 기술>

```
module psc(clk_1x, clk_2x, rst, x1, x2, y);
    parameter N = 8;
    input clk_1x, clk_2x, rst;
    input [N-1:0] x1, x2;
    output reg [N-1:0] y;

    reg [2*N-1:0] q;
    reg sel;

    always @(posedge clk_1x) begin // slow clk
        if (rst) q <= 0;
        else q <= {x1, x2};
    end

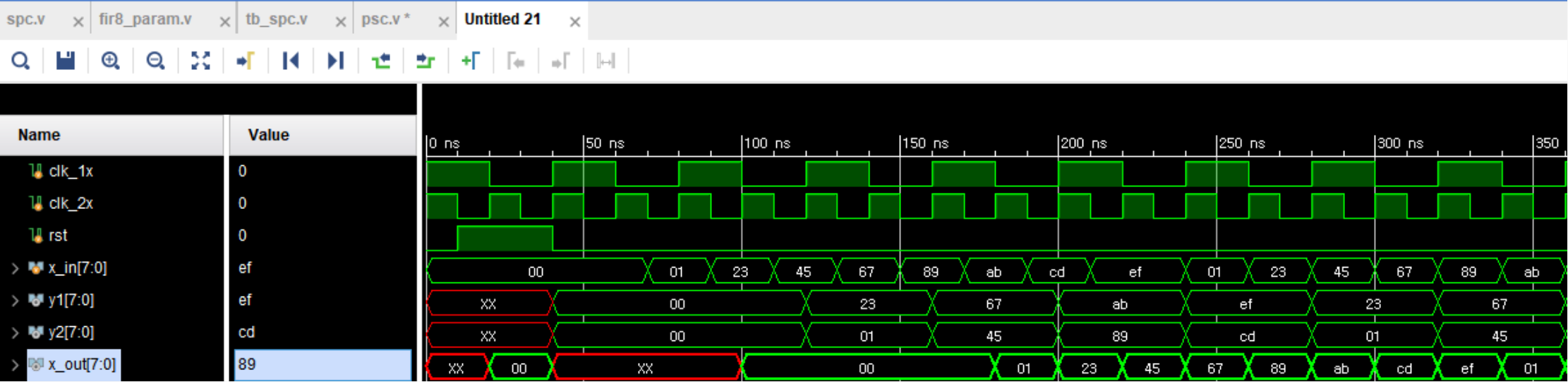
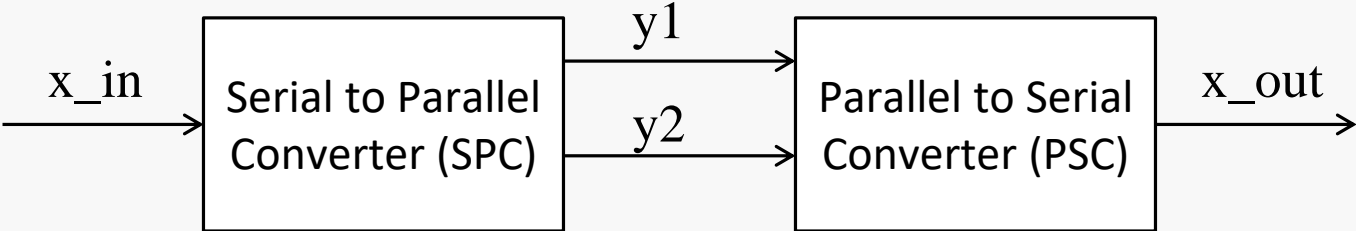
    always @(posedge clk_2x) begin
        if (rst) sel <= 0;
        else sel <= ~sel;
    end

    always @(posedge clk_2x) begin // fast clk
        if (rst) y <= 0;
        else if (sel) y <= q[N-1:0];
        else y <= q[2*N-1:N];
    end

endmodule
```

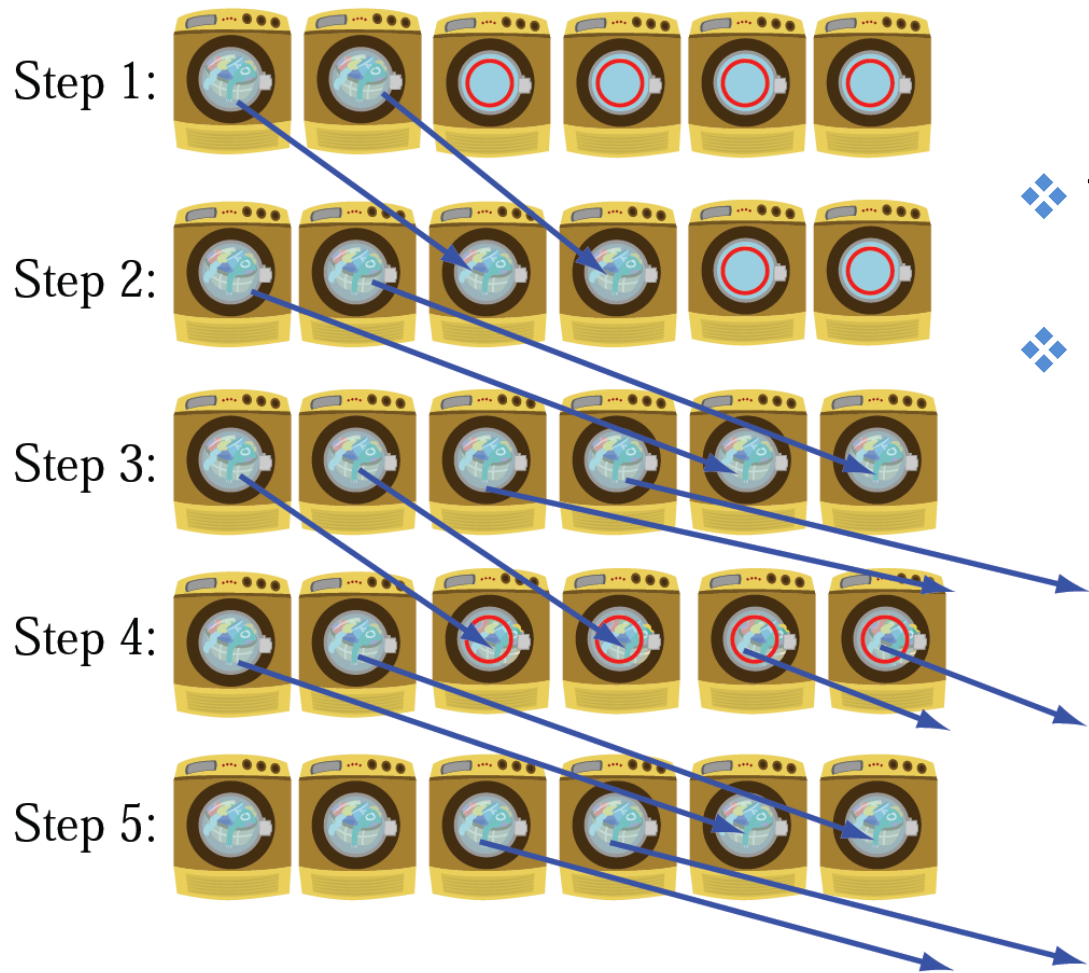
4.4. 유한 임펄스 응답 필터의 가속 설계 방법

◆ SPC-PSC의 연결을 위한 테스트 벤치



4.4. 유한 임펄스 응답 필터의 가속 설계 방법

◆ 병렬 처리 (parallel processing ($\times N$))



- ❖ Throughput = $\frac{2}{30} = \frac{1}{15}$ load/min
- ❖ Latency = 90 mins

Q & A

Q&A

THANK YOU -

경청해 주셔서 감사합니다.

