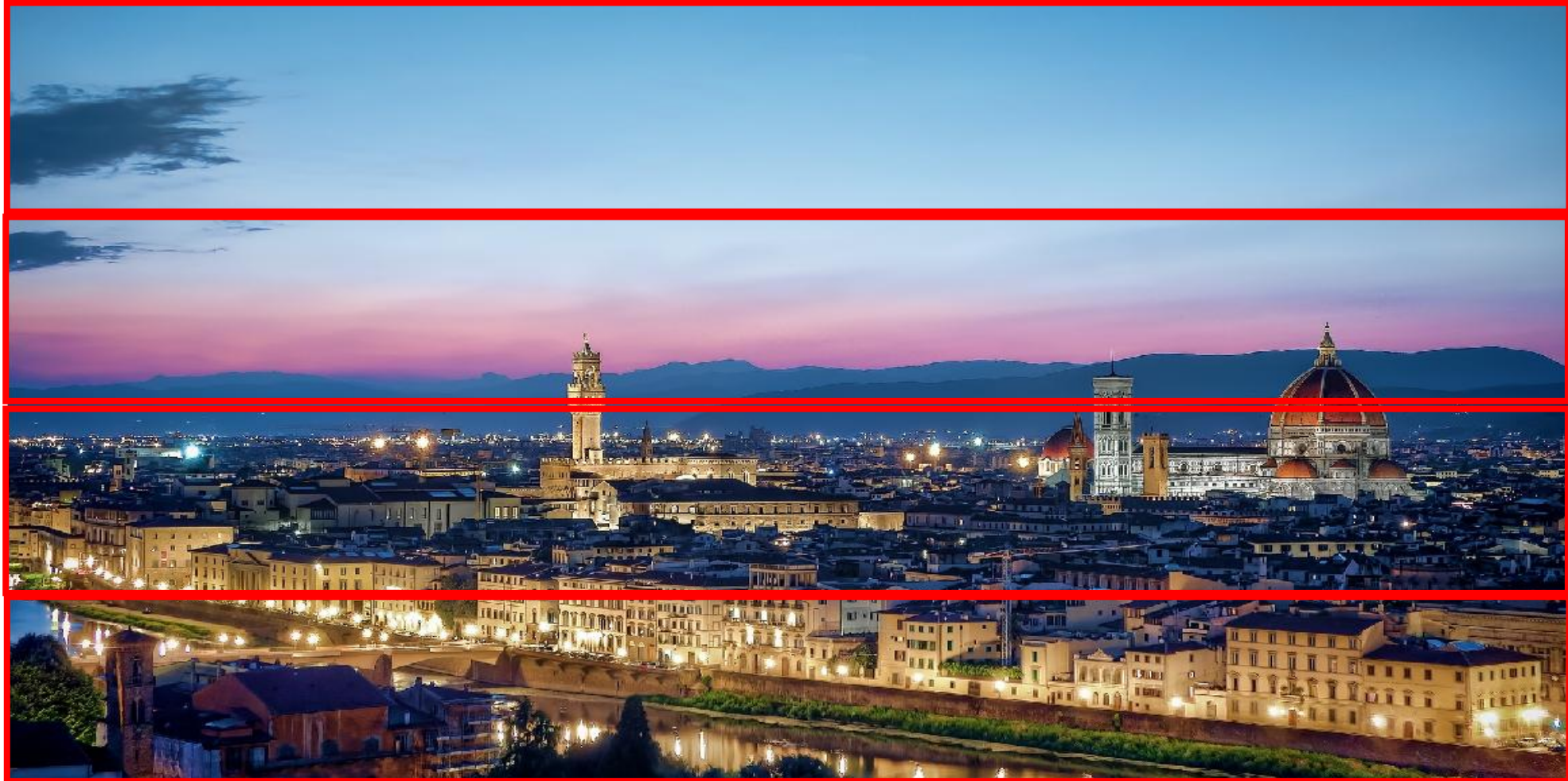


# 최종발표

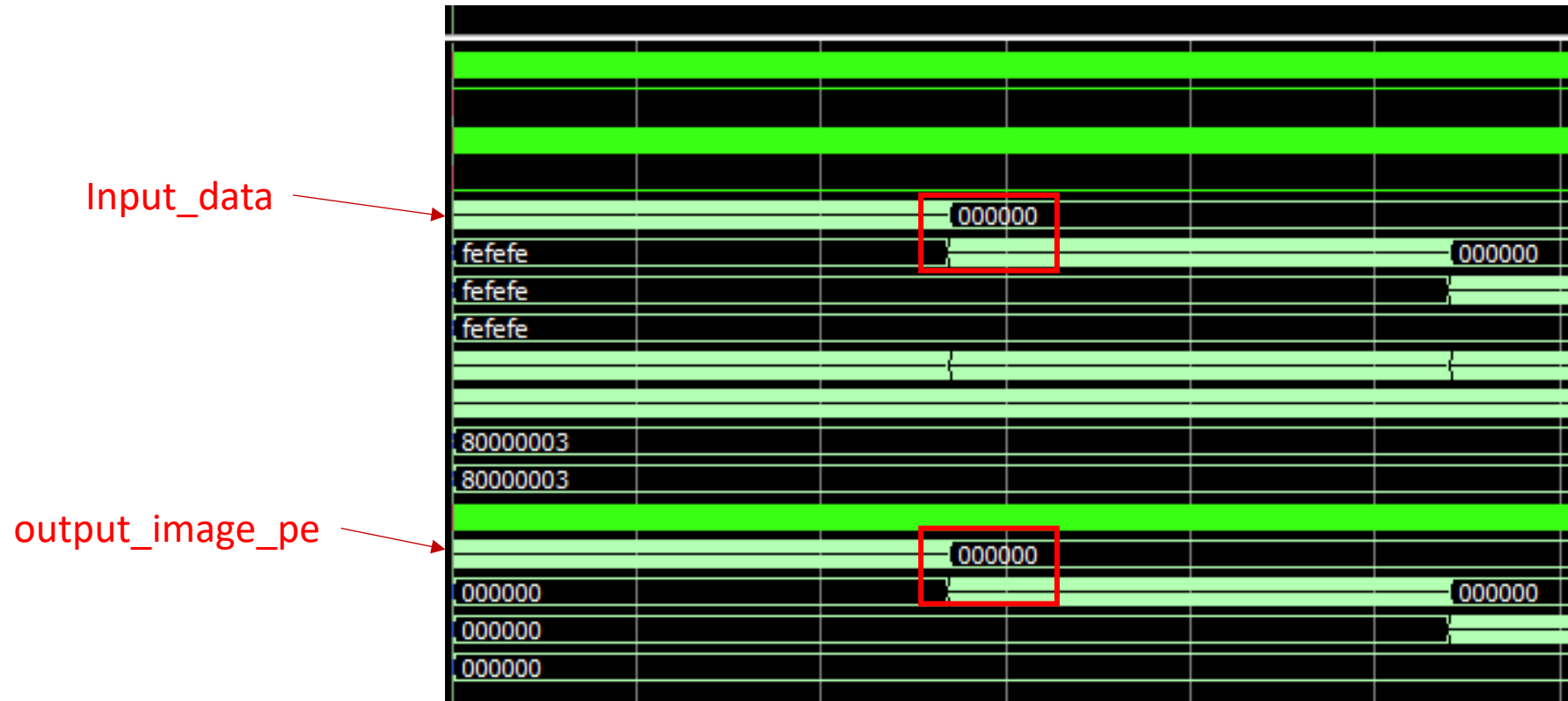
5조

조장: 송0호  
팀원: 김0진, 이0백, 한0윤

이미지를 4등분 (1920X270)하고 각 부분을 PE가 병렬적으로 처리

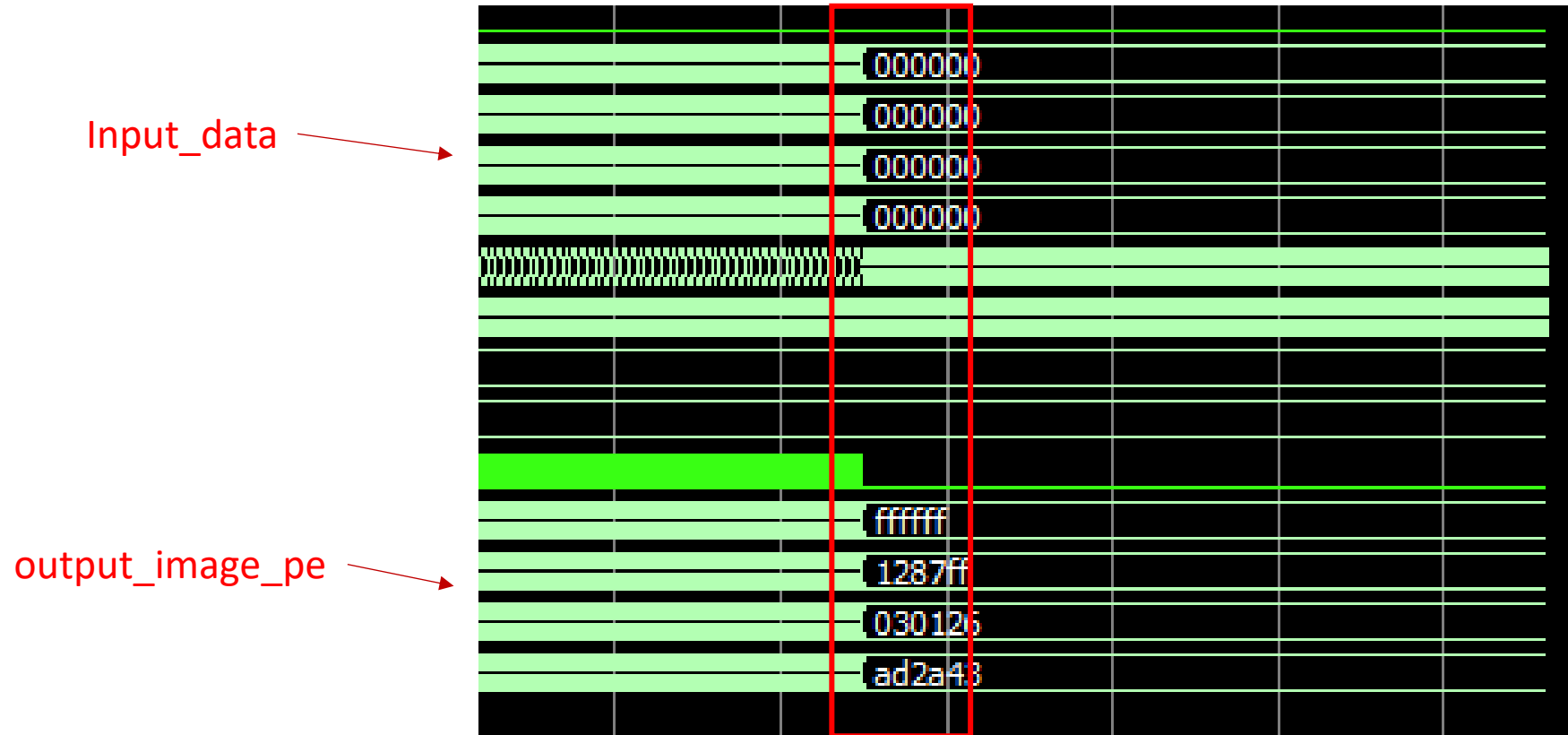


오류: 별도의 4개의 for문



한 개의 PE 종료 후 다른 한 개의 PE 실행

수정: 최종 이미지 병합 과정

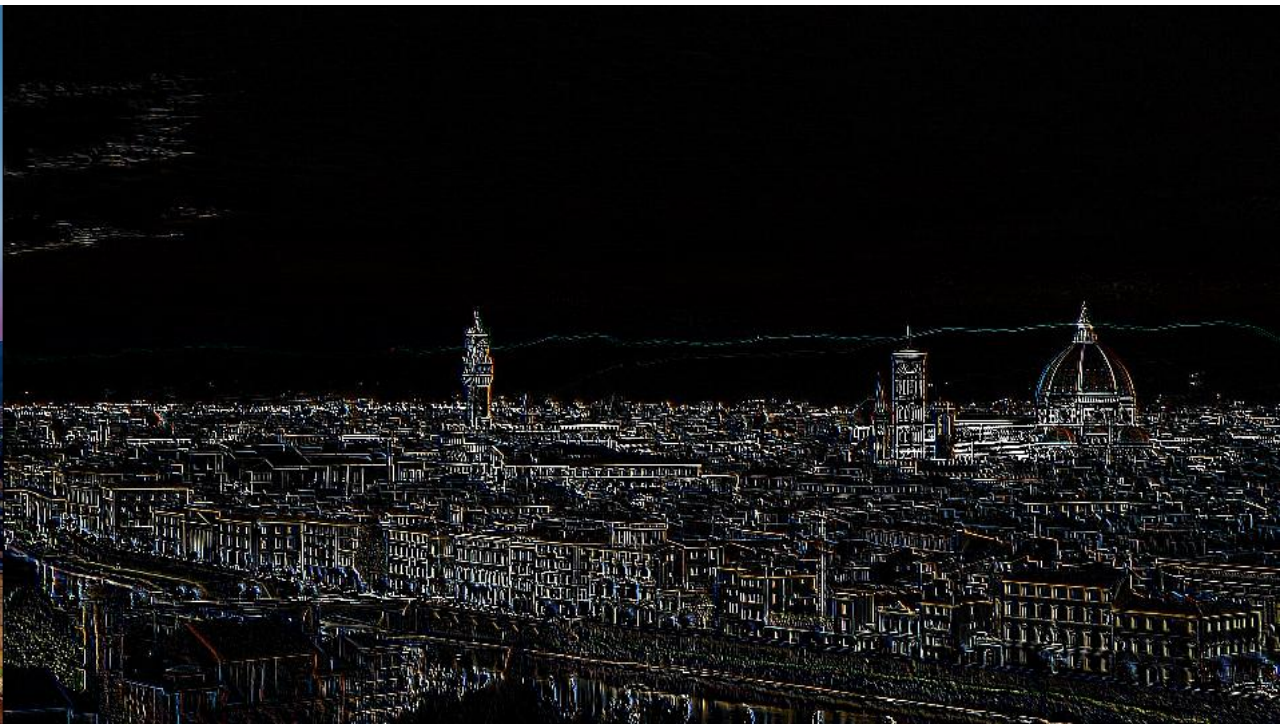


4개의 PE 동시 동작, **BUT** 최종 이미지 병합 시간 소요



# 1920X270 filter – 4PE

중간 이미지 reg 삭제 후 최종 이미지 출력



Now	456192360 ns
Cursor 1	4225779 ns

<PE 1개>

Now	114048360 ns
Cursor 1	4225779 ns

<PE 4개>

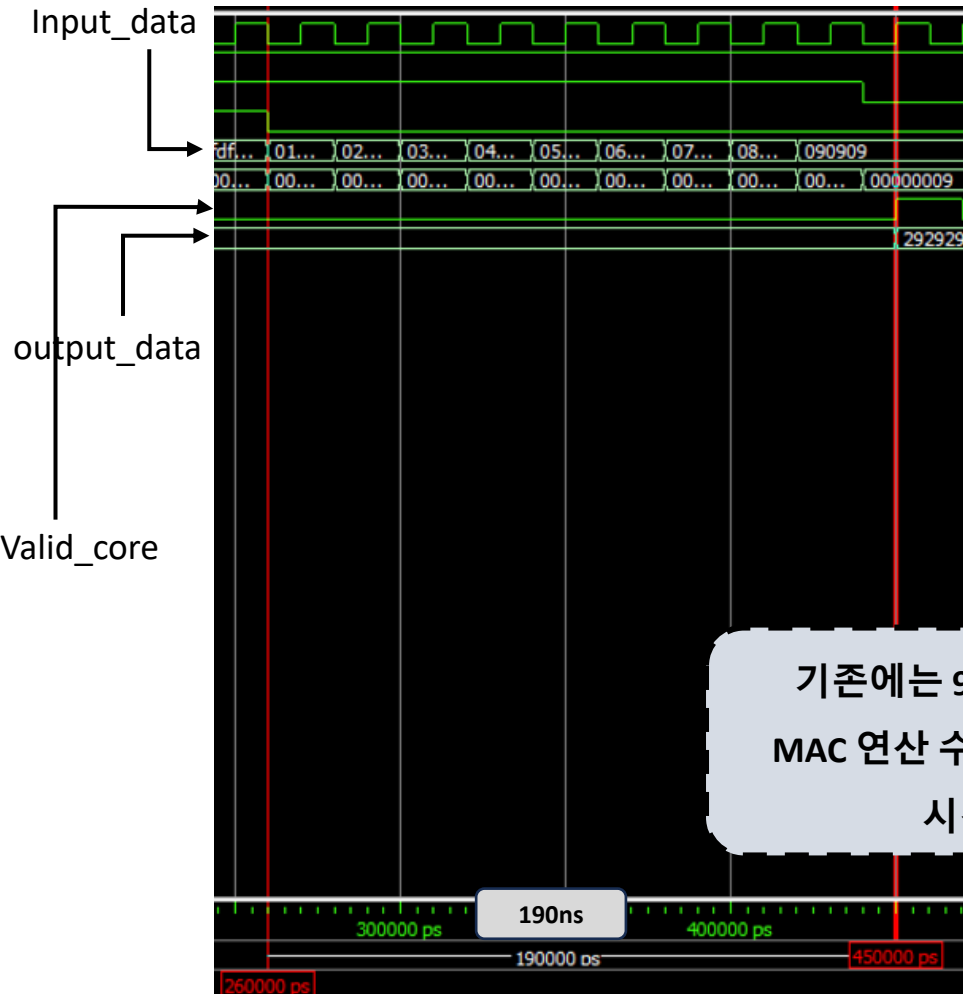
$456,192,360 / 114,048,360 = 3.999$  약 4배

최종 이미지 정상 출력 & 목표 4배 달성

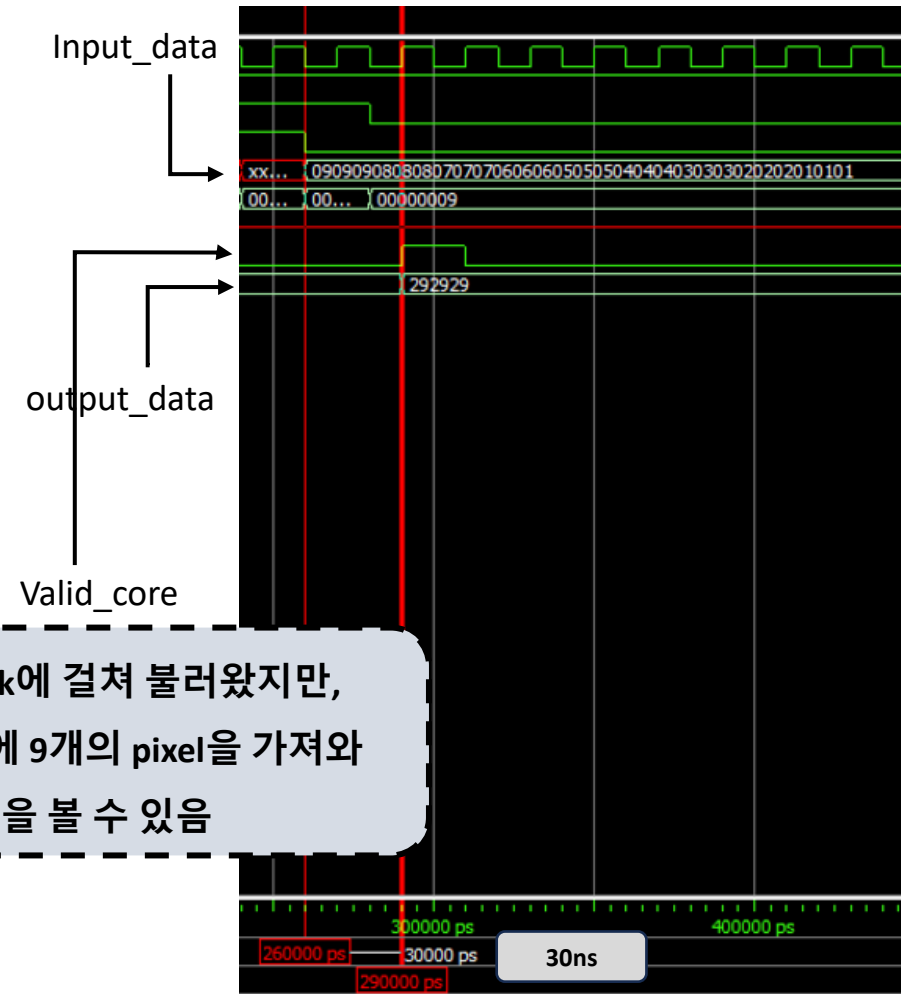
# MAC 연산을 통한 성능 개선

## 3x3 이미지 필터링에서의 MAC 연산 개선

< 기존 3x3 이미지 필터링 파형 >



< 3x3 이미지 필터링 MAC 수정 시 파형 >

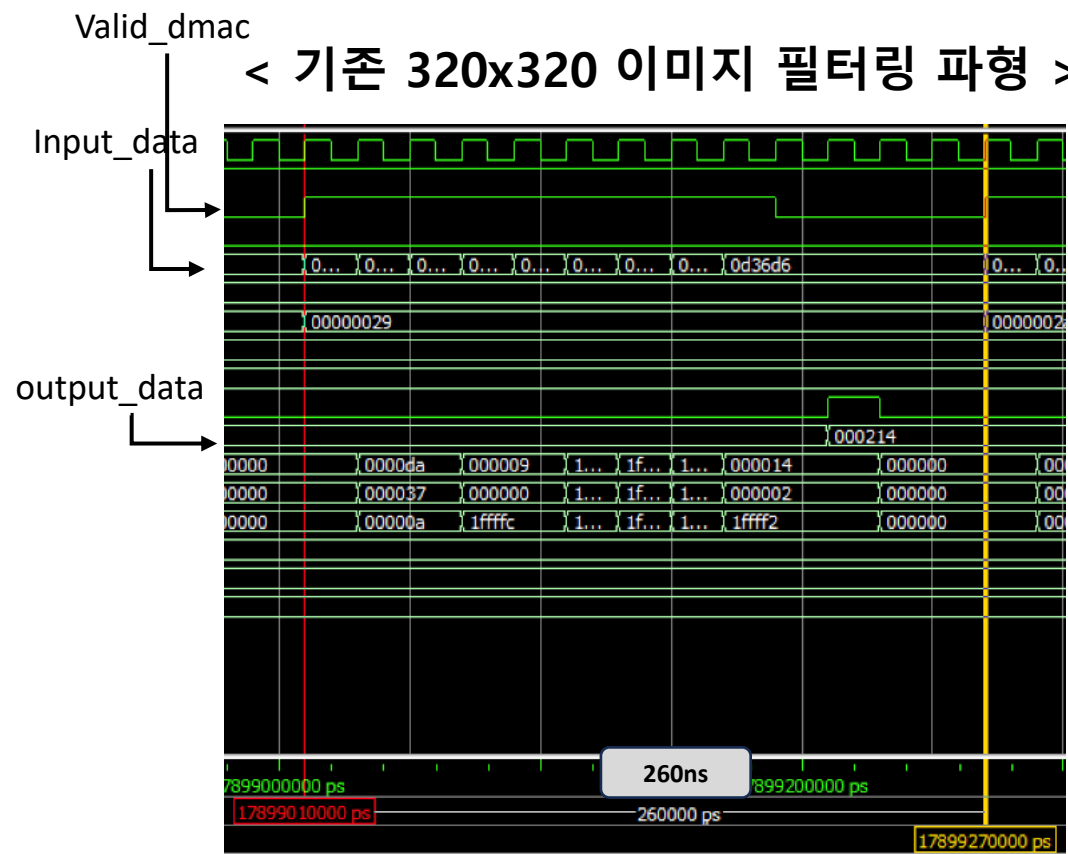


기존에는 9개의 pixel을 9 clk에 걸쳐 불러왔지만,  
MAC 연산 수정을 통해 1 clk에 9개의 pixel을 가져와  
시간적으로 단축함을 볼 수 있음

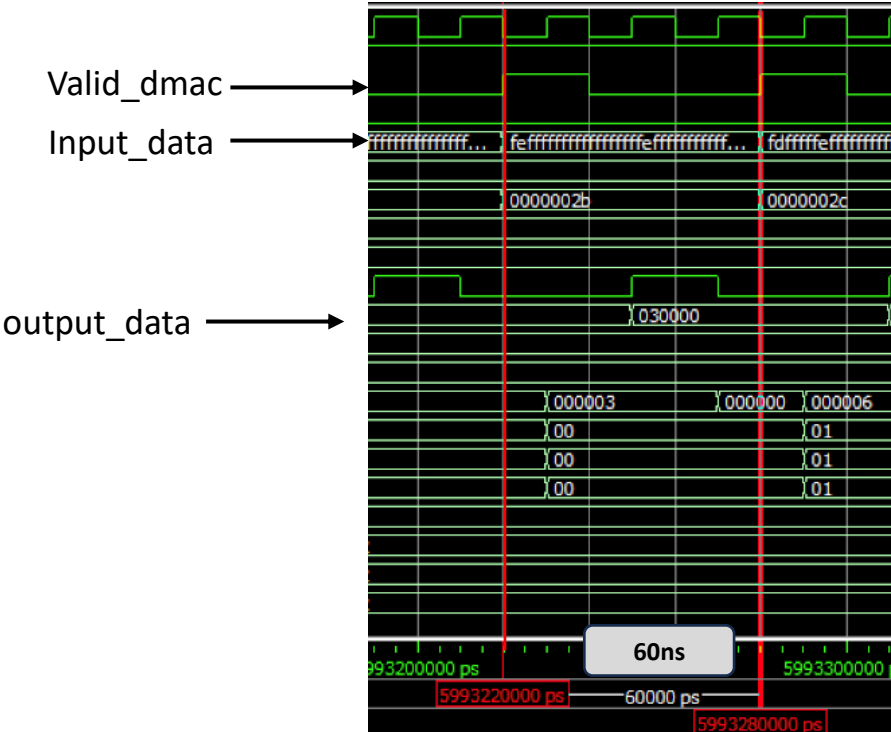
# MAC 연산을 통한 성능 개선

## 320x320 이미지 필터링에서의 MAC 연산 개선

< 기존 320x320 이미지 필터링 파형 >



< 320x320 이미지 필터링 MAC 수정 시 파형 >



기존 9개의 pixel에 대한 이미지 필터링에 걸리는 시간: 260ns

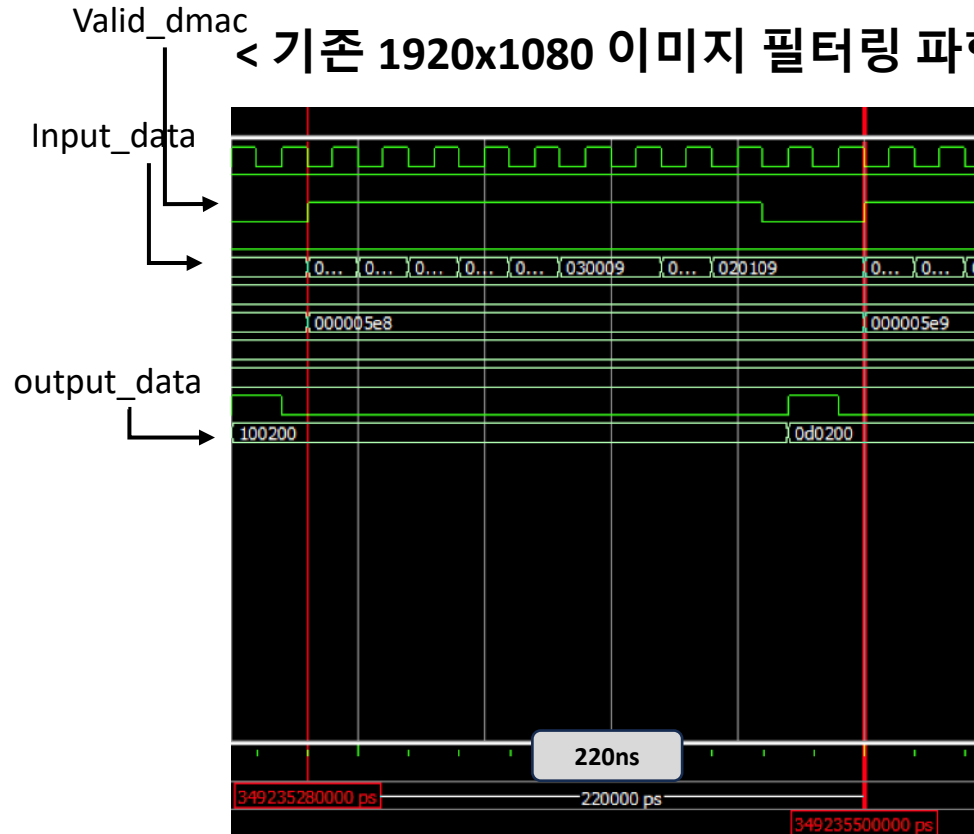
MAC 수정 후 9개의 pixel에 대한 이미지 필터링에 걸리는 시간: 60ns

약 4배의 속도 개선

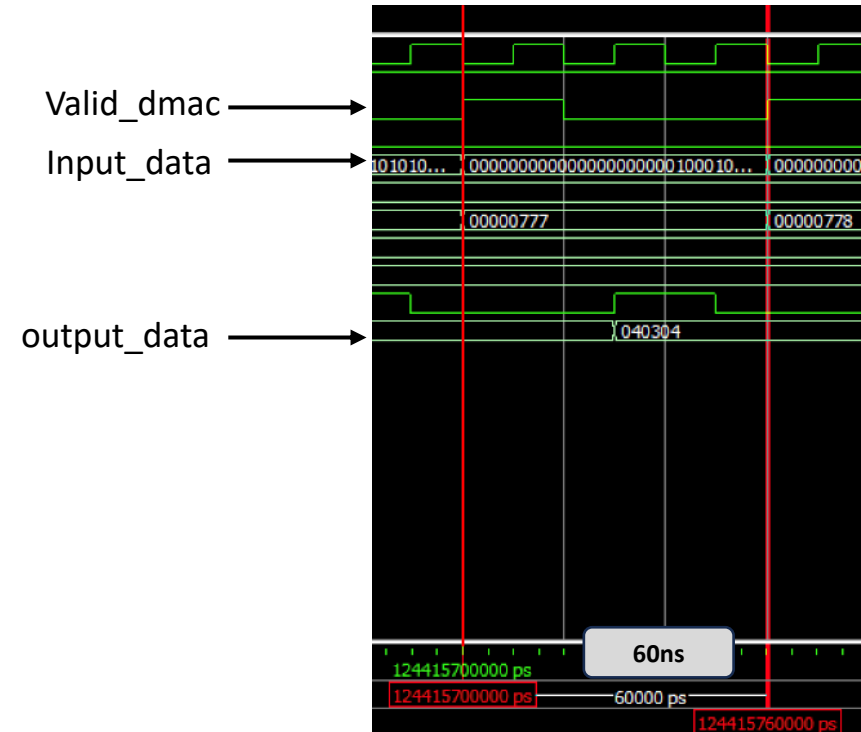
# MAC 연산을 통한 성능 개선

## 1920x1080 이미지 필터링에서의 MAC 연산 개선

< 기존 1920x1080 이미지 필터링 파형 >



< 1920x1080 이미지 필터링 MAC 수정 시 파형 >



기존 9개의 pixel에 대한 이미지 필터링에 걸리는 시간: 260ns  
MAC 수정 후 9개의 pixel에 대한 이미지 필터링에 걸리는 시간: 60ns  
3.66배의 속도 개선




# MAC 연산을 통한 성능 개선

1920x1080 이미지 필터링에서의 MAC 연산 개선

< 기존 1920x1080 최종 이미지 출력까지 걸린 시간 >

Now	456192360000 ps
Cursor 1	349235280000 ps
Cursor 2	349235500000 ps
456,192,360ns	

< 1920x1080 MAC 수정 적용하여 최종 이미지까지 걸린 시간 >

 Now	124416360000 ps
Cursor 1	124415700000 ps
Cursor 2	124415760000 ps
124,416,360ns	

$456,192,360\text{ns} / 124,416,360\text{ns} = 3.66\text{배}$  속도 개선

# MAC 연산을 통한 성능 개선

## 진행 과정 중 오류

### 오류1:

모듈의 input\_data로 array 전달 불가능

```
...fir_filter_2d_tb.v -- Unsuccessful Compile
vlog -work work -vopt -stats=none C:/Users/sean/Digital_system_hws/MAC_change_testbench_1_3x3/fir_filter_2d_tb.v
Questa Intel Starter FPGA Edition-64 vlog 2021.2 Compiler 2021.04 Apr 14 2021
-- Compiling module fir_filter_2d_tb
** Error: C:/Users/sean/Digital_system_hws/MAC_change_testbench_1_3x3/fir_filter_2d_tb.v(35): (vlog-2110) Illegal reference to memory "input_data".
** Error: C:/Users/sean/Digital_system_hws/MAC_change_testbench_1_3x3/fir_filter_2d_tb.v(35): (vlog-2110) Illegal reference to net array "#implicit-wire#2".
** Error: C:/Users/sean/Digital_system_hws/MAC_change_testbench_1_3x3/fir_filter_2d_tb.v(35): (vlog-2110) Illegal reference to memory "input_data".
```

### 오류2:

Part-select 시 변수 불가능

```
...fir_filter_2d_tb.v -- Unsuccessful Compile
vlog -work work -vopt -stats=none {C:\Users\sean\Digital_system_hws\MAC_change_testbench_1_3x3\fir_filter_2d_tb.v}
Questa Intel Starter FPGA Edition-64 vlog 2021.2 Compiler 2021.04 Apr 14 2021
-- Compiling module fir_filter_2d_tb
** Error: C:\Users\sean\Digital_system_hws\MAC_change_testbench_1_3x3\fir_filter_2d_tb.v(61): Range width must be constant expression.
```

# MAC연산 처리 방법 변경 + PE개수 증가에 따른 성능 향상 비율

a) MAC연산 변경 \* PE 1개 사용 - 124,416,360ns (3.66배 감소)

Now	124416360 ns
Cursor 1	1152660 ns

시뮬레이션 시간

00:03:12.59

현실 타이머 시간

b) MAC연산 변경 \* PE 2개 사용 - 62,208,360ns (약 7.33배( $3.66 \times 2$ ) 감소)

Now	62208360 ns
Cursor 1	1152660 ns

시뮬레이션 시간

00:02:40.33

현실 타이머 시간

# MAC연산 처리 방법 변경 + PE개수 증가에 따른 성능 향상 비율

c) MAC연산 변경 \* PE 3개 사용 – 41,472,360ns (약 11배( $3.66*3$ ) 감소)

Now	41472360 ns	00:02:29.98
Cursor 1	1152660 ns	

시뮬레이션 시간

현실 타이머 시간

d) MAC연산 변경 \* PE 4개 사용 – 31,104,360ns (약 14.64배( $3.66*4$ ) 감소)

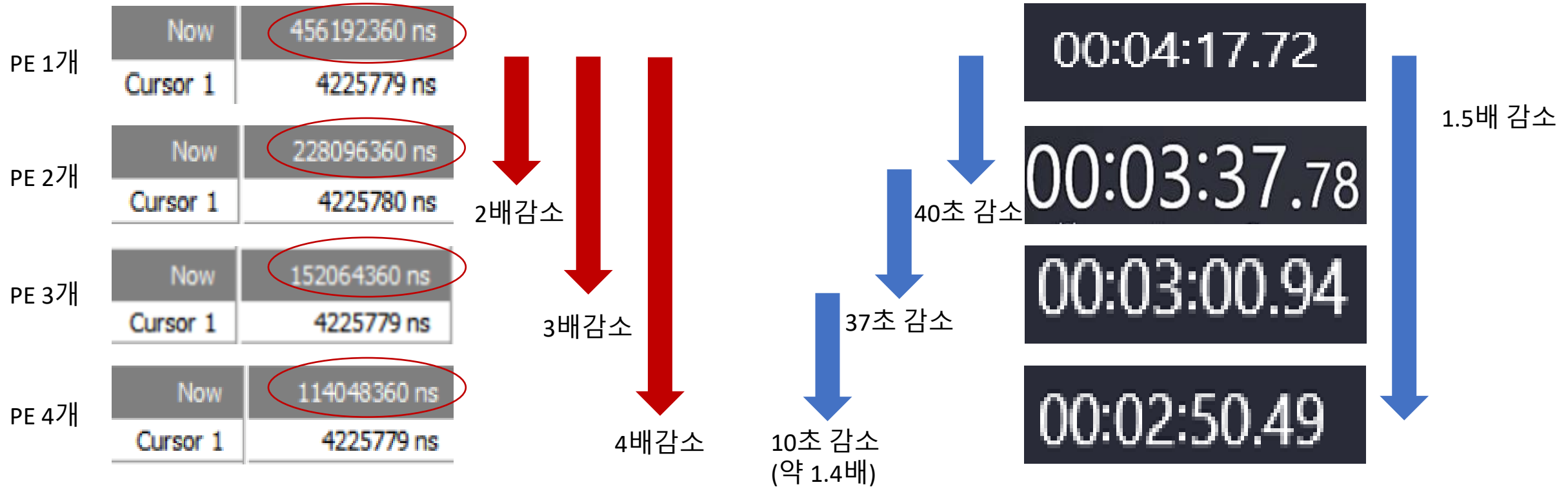
Now	31104360 ns	00:02:18.69
Cursor 1	1152660 ns	

시뮬레이션 시간

현실 타이머 시간

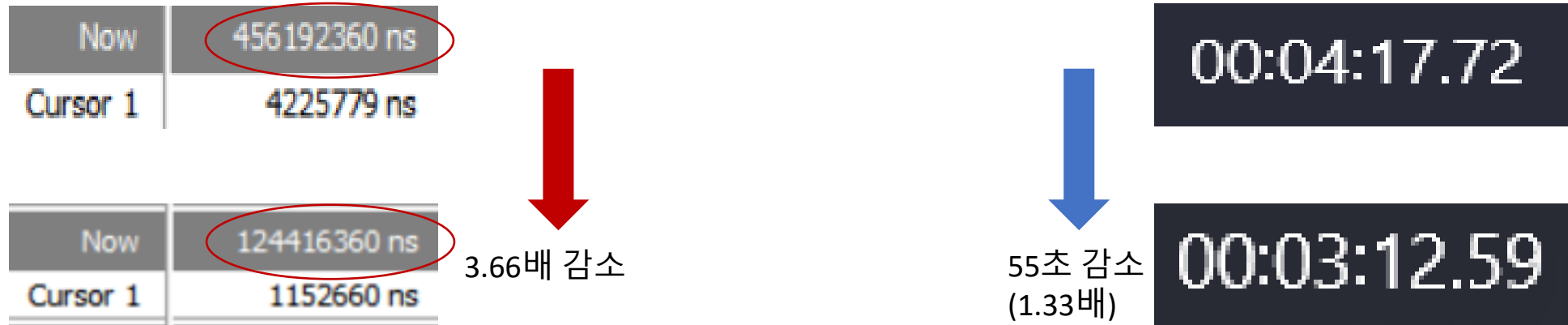
# 결과 분석

PE를 4개로 설정한 이유



- PE를 늘릴수록 시뮬레이션 시간은 배수적으로 감소 // 현실 시간은 감소폭이 점점 줄어듦
- PE개수를 늘림에 따라 회로에서 생기는 reg/wire의 개수 증가 → delay증가
- 감소하는 시간보다 증가하는 소자에 대한 비효율성이 크다고 판단

## MAC 연산에서 시간 비교

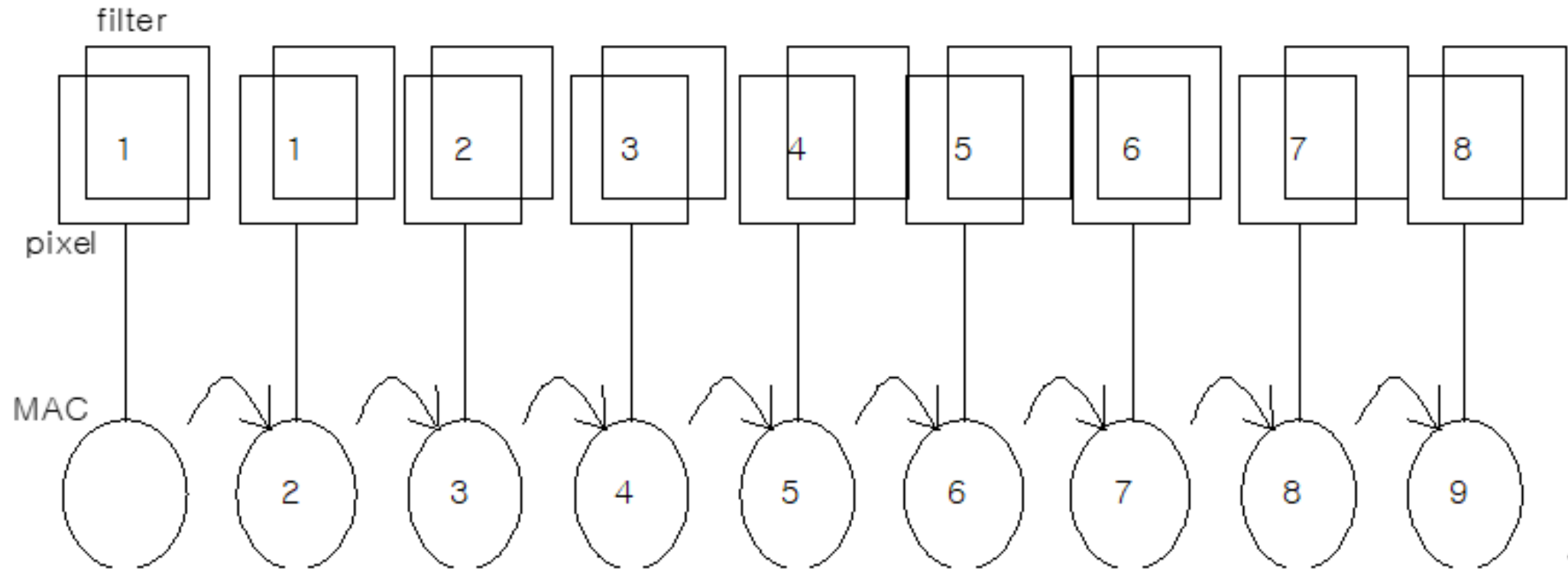


-수정 전 MAC에서는 9번의 곱 연산, 8번의 합 연산을 9타임에 걸쳐서 수행

- 수정 후 MAC에서는 9번의 곱 연산, 8번의 합 연산을 5타임에 걸쳐서 수행 →  $9/5=1.8$ 배의 속도 감소

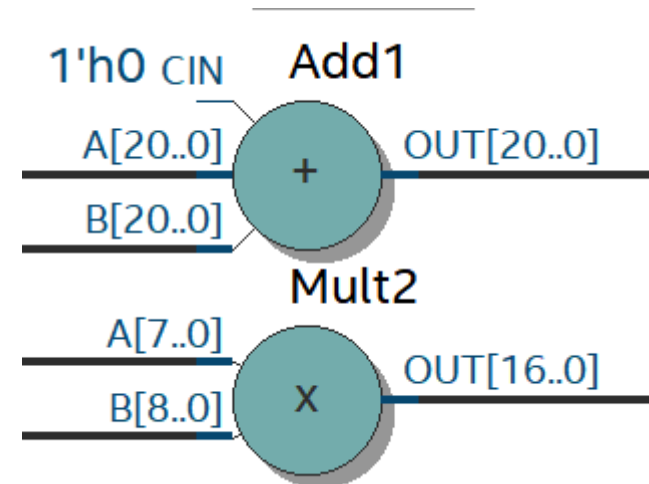
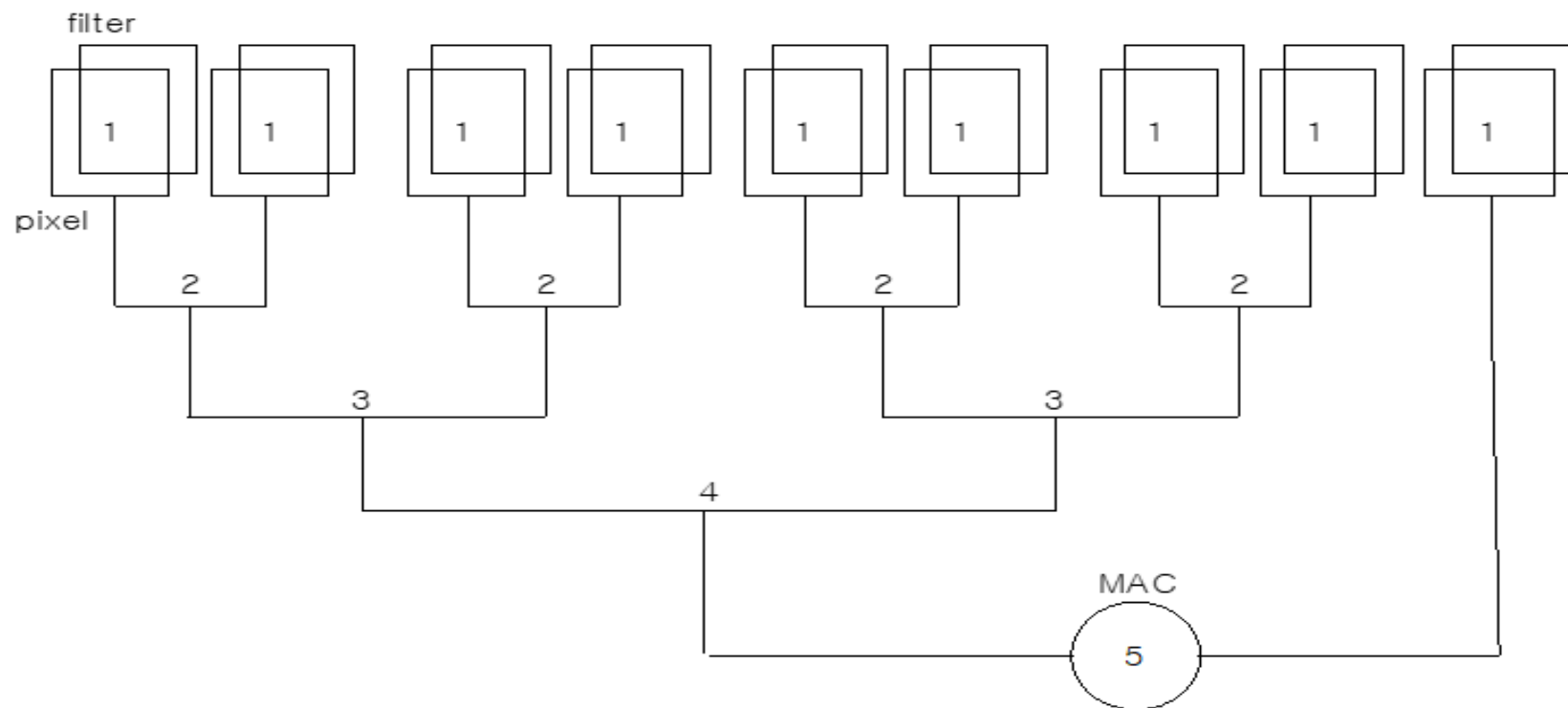


## MAC 연산에서 시간 비교



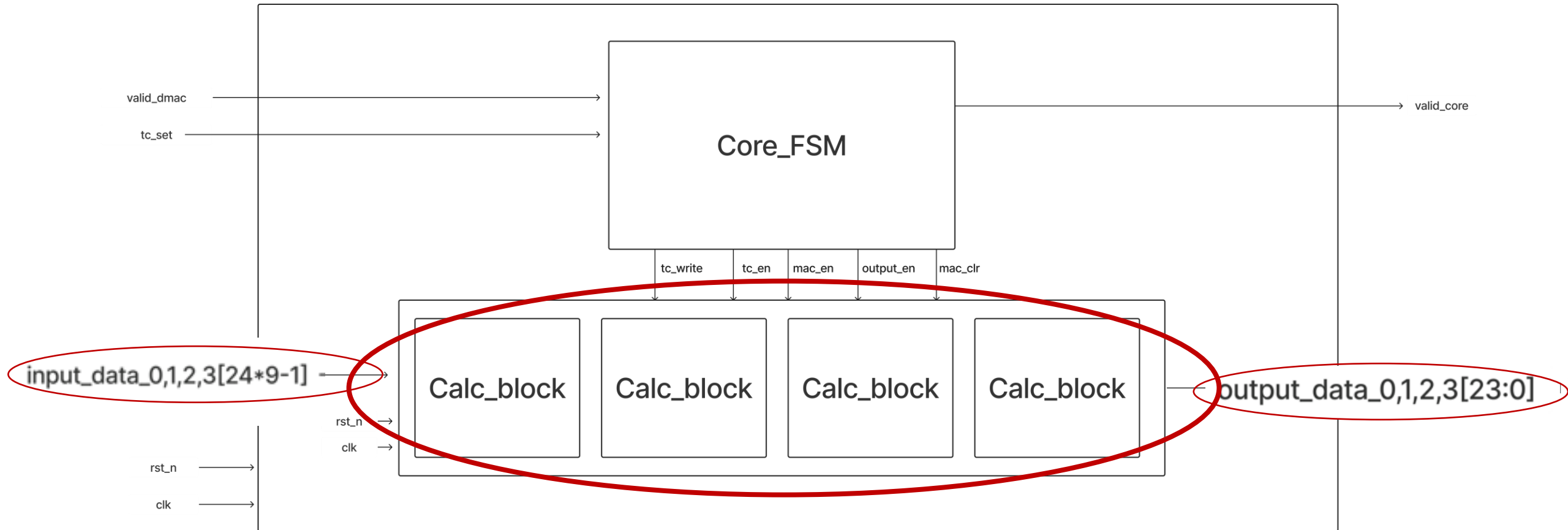
- 수정 전 MAC에서는 9번의 곱 연산, 8번의 합 연산을 9타임에 걸쳐서 수행
- 시뮬레이션 상에서는 9cycle에 걸쳐 수행

## MAC 연산에서 시간 비교



- 수정 후 MAC에서는 9번의 곱 연산, 8번의 합 연산을 5타임에 걸쳐서 수행
- 시뮬레이션 상에서는 1cycle에 걸쳐 수행

## PE4개 병렬처리 \* MAC 수정



- 최종적으로 4개의 PE사용 (4배 감소) \* MAC 연산의 수정 (3.66배 감소) = 총 14.64배의 시간 감소