

Lecture 11~12

영상처리

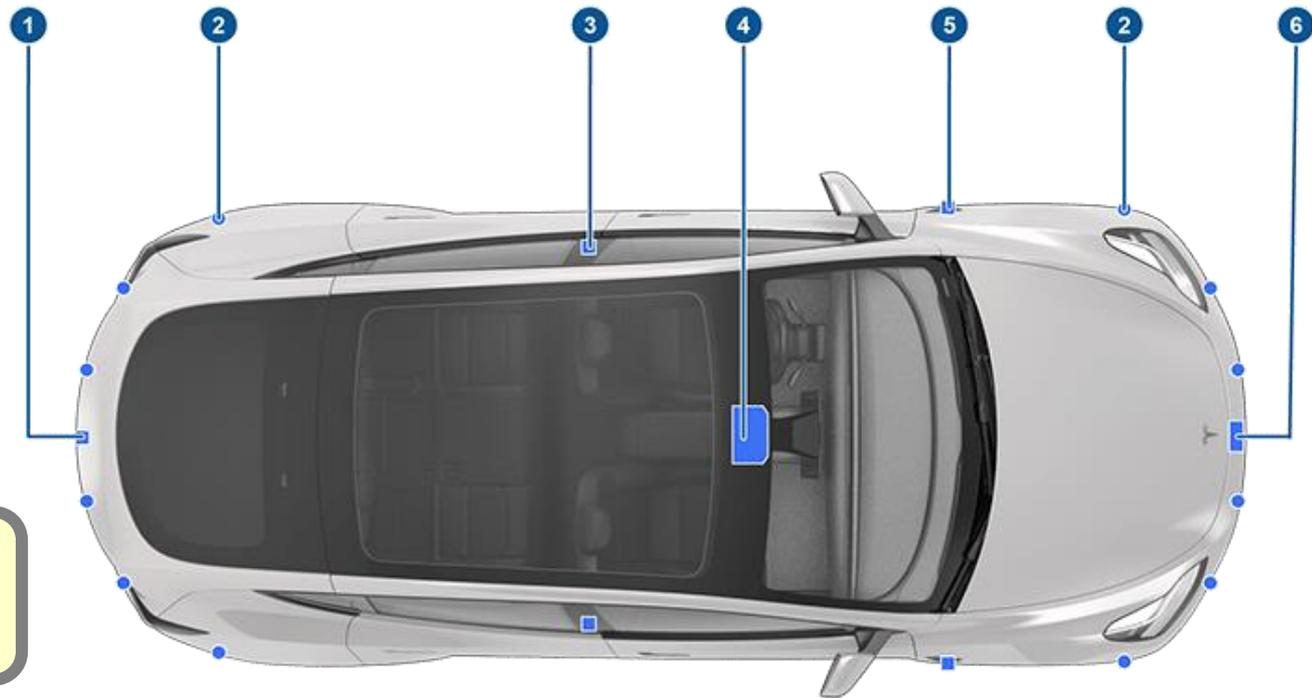
비전 기반 시스템

■ 자율주행차(Tesla)

1. 후방 카메라
2. 초음파 센서
3. 카메라
4. 카메라(3개)
5. 카메라
6. 레이더(radar)



카메라 시스템 기반
자율주행



비전 기반 시스템



- 스마트 CCTV 시스템
 - 얼굴인식, 객체인식, 숫자인식(번호판 등), 이상행위탐지, ...

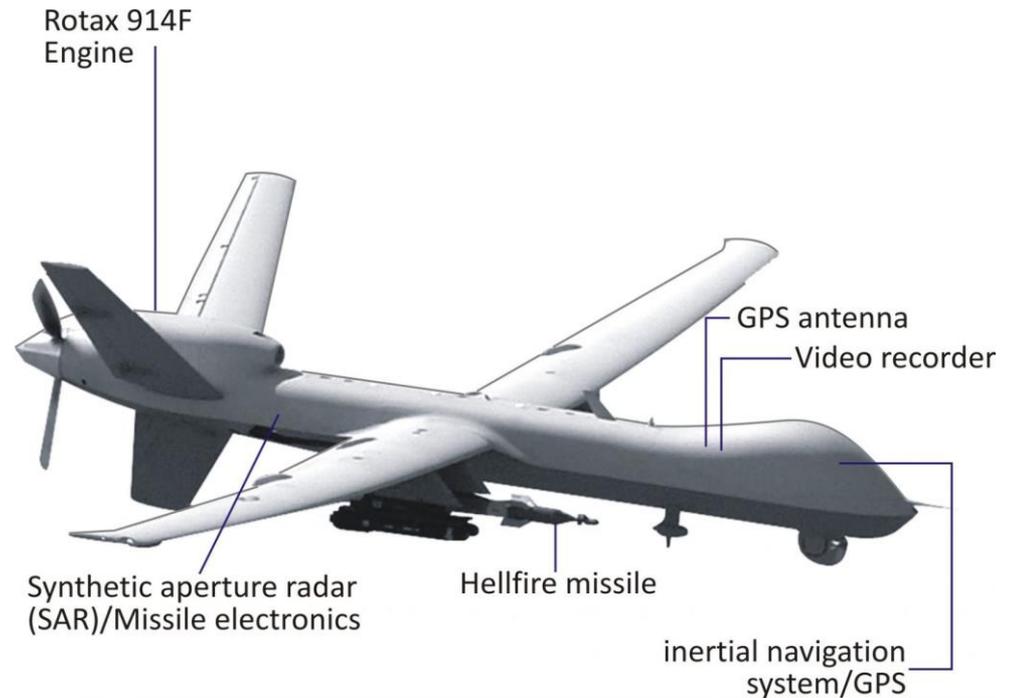


비전 기반 시스템



- 드론 및 무인항공기(UAV)

- 응급구조서비스, 농업(가축 모니터링, pest and disease control, ...), 건설, ...



디지털 카메라



Multi purpose camera

Combination of classic cutting edge computer vision algorithms and artificial intelligence methods

Safe perception through an algorithmic multi-path approach



large field of view

for detection of crossing vulnerable road users

high angular resolution

with increased range at the center

artificial intelligence

for robust perception and behavior prediction



TECHNICAL CHARACTERISTICS

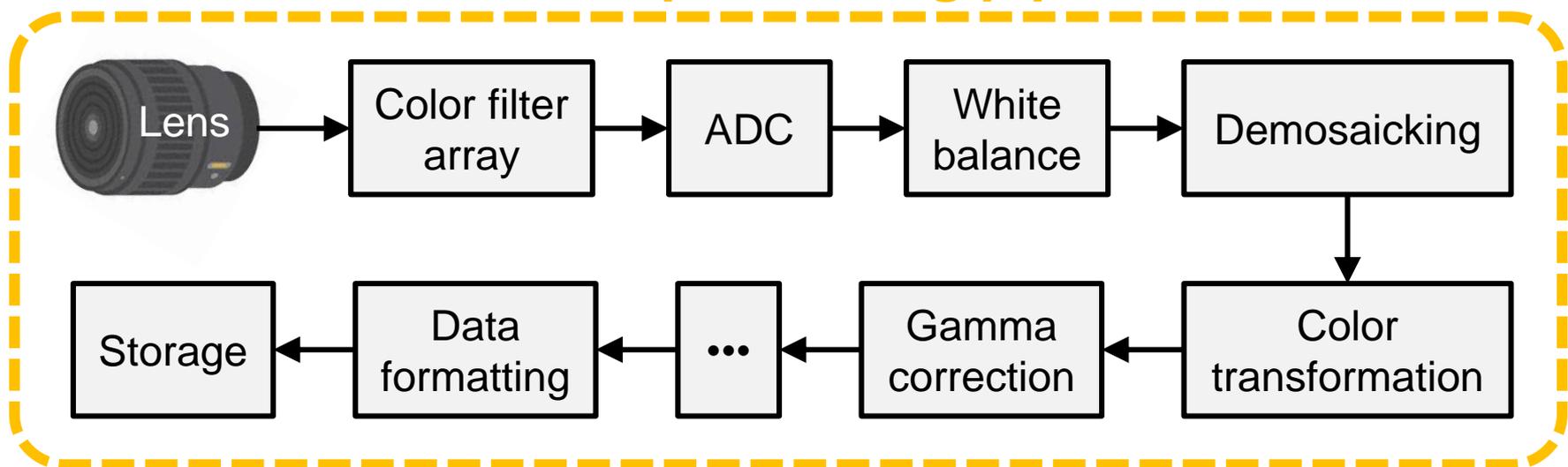
Optics	Horizontal field of view	$\pm 50^\circ$
	Vertical field of view	27° up, 21° down
	Aperture	F1.8
Imager	Resolution	2.6 MP HDR (2,048 x 1,280 pixels)
	Color pattern	RCCG
	Frame rate	45 frames per second with flicker mitigation
System on chip	Technology	16 nm FFC
	Processing system	4 x ARM quad core (~ 9000 DMIPS) + 1 x ARM dual lockstep
	Hardware accelerator	DNN, classifier, optical flow, flexible CV engines
Safety level		Up to ASIL-B
Mechanics	Box size	120 x 61 x 36 mm

- ▶ System-on-chip with Bosch know-how for cutting-edge algorithm performance
- ▶ Reliable full scene understanding for increased safety using algorithmic multi-path approach
- ▶ Semantic segmentation based on deep learning
- ▶ Optical flow for model-free video processing

디지털 카메라

- 이미지 센서 + ISP(Image Signal Processor)
- 출력 데이터 포맷
 - 이미지 픽셀: MONO8/12, RGB8/16, YUV 4:2:2, ...
 - 동기화 신호: horizontal/vertical active video signals

In-camera processing pipeline



이미지 데이터 포맷

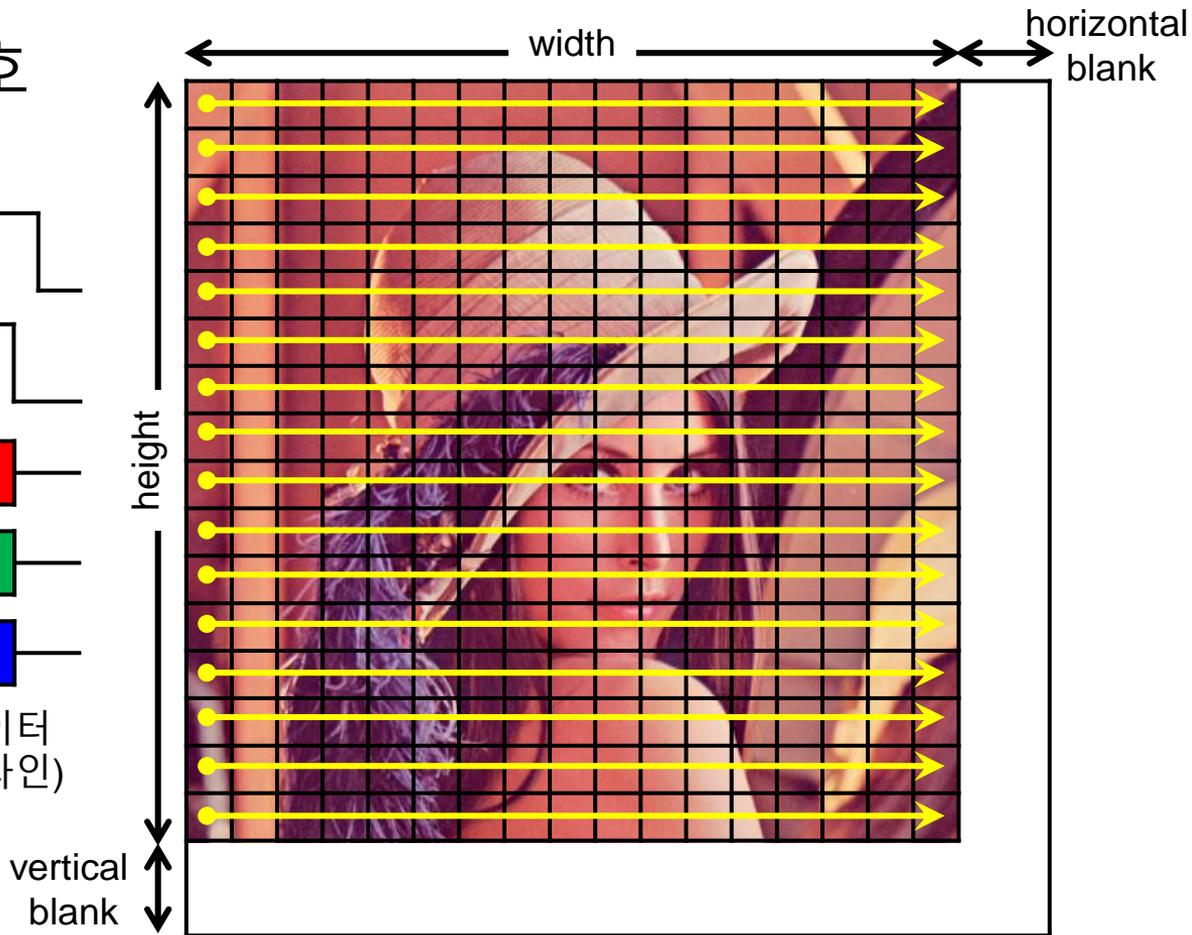
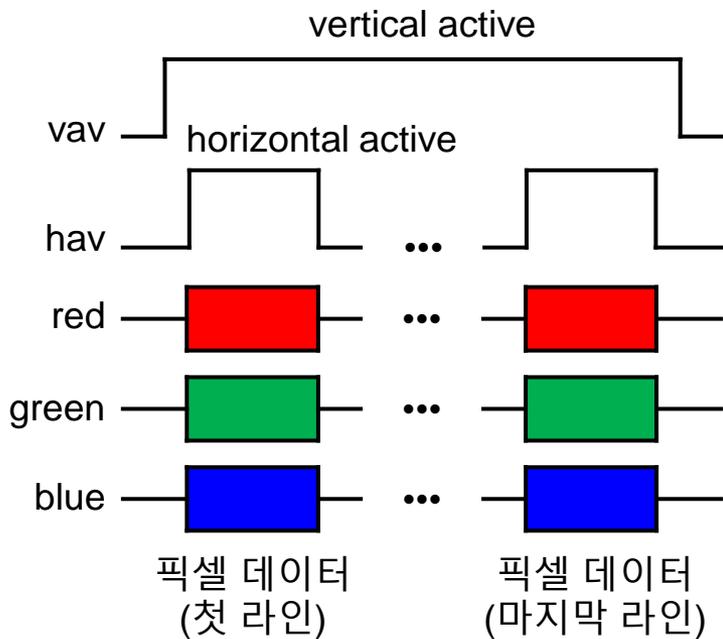
- RGB 이미지



				0				1				2			
				channel											
100	103	250	22	5	47										
80	30	232	133	5	45										
122	39	211	152												0
113	56	0	120												65
				20	50	1	105								0
								125	127	213	48				

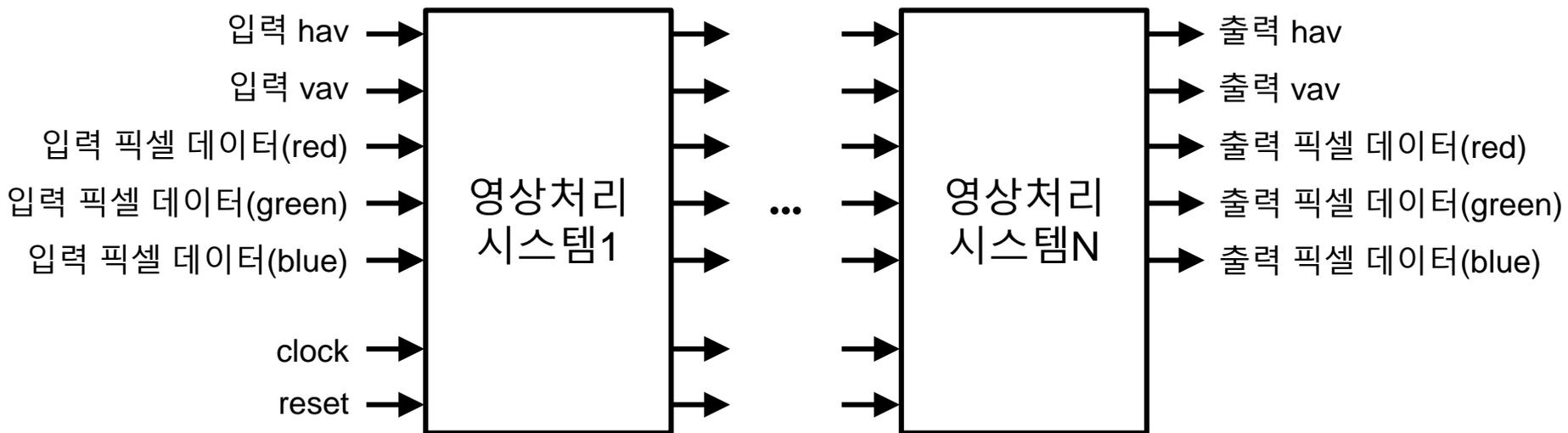
이미지 데이터 포맷

픽셀 및 동기화 신호



영상처리 시스템

- 입력
 - 픽셀 데이터 및 동기화 신호
 - 클럭 및 리셋 신호
- 출력
 - 픽셀 데이터 및 동기화 신호



실습1: RGB to Gray

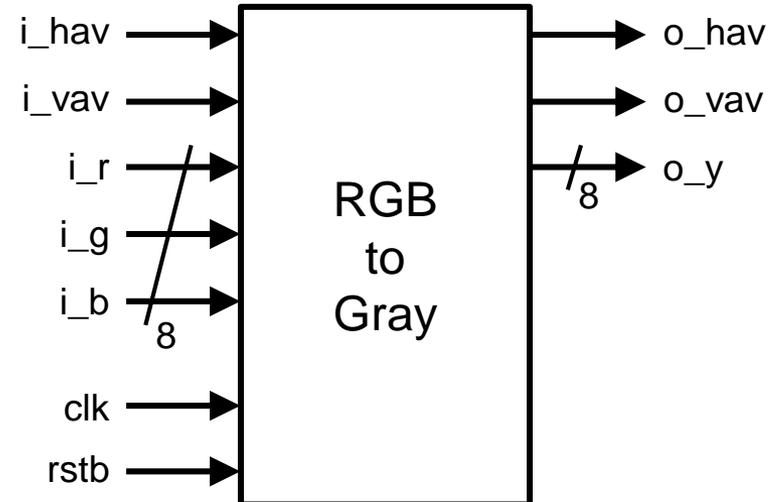
- 입력
 - 8비트 픽셀 데이터 : red, green, blue
 - 동기화 신호 : hav, vav
 - 클럭 및 리셋 신호 : clk, rstb

- 출력
 - 8비트 픽셀 데이터 : gray
 - 동기화 신호 : hav, vav

- 변환 공식

$$Y = 0.299R + 0.587G + 0.114B$$

0.299, 0.587 및 0.144을
2진수로 어떻게 표현???



실습1: RGB to Gray



- 2진수로 분수 표현 방식
 - 원하는 비트 크기 결정
 - 8비트
 - 2진수 변환 : $\text{binary} = \text{round}(\text{분수} * 2^{\text{비트크기}})$
 - $0.299 \rightarrow \text{round}(0.299 * 2^8) = 77$, 오차 = $\left|0.299 - \frac{77}{2^8}\right| = 0.00178125$
 - $0.587 \rightarrow \text{round}(0.587 * 2^8) = 150$, 오차 = $\left|0.587 - \frac{150}{2^8}\right| = 0.0010625$
 - $0.114 \rightarrow \text{round}(0.114 * 2^8) = 29$, 오차 = $\left|0.114 - \frac{29}{2^8}\right| = 0.00071875$
 - 오차 평가
 - 8비트 사용 $\rightarrow 1\text{LSB} = 2^{-8} = 0.00390625$
 - 오차가 $\pm 1\text{LSB}$ 범위에 머무름

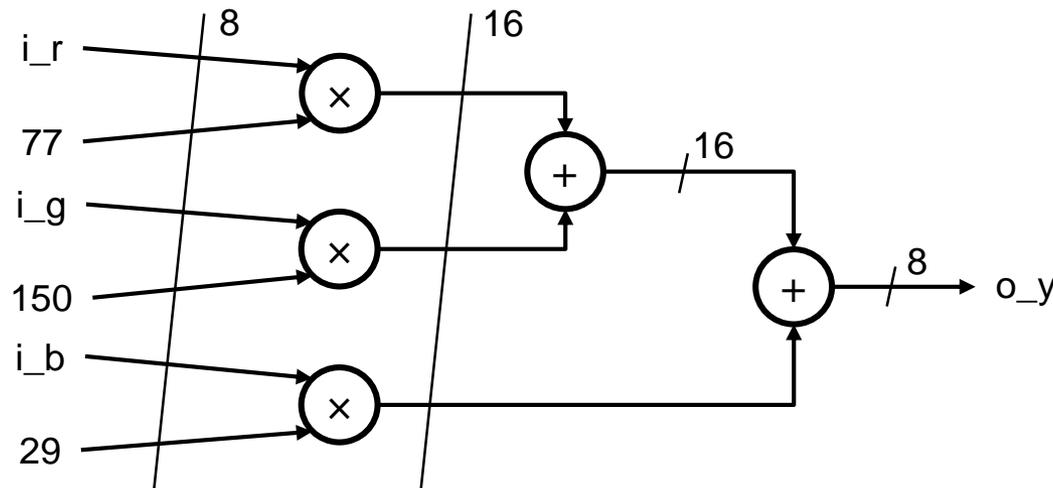
실습1: RGB to Gray

- RGB to Gray 변환 공식

$$Y = 0.299R + 0.587G + 0.114B$$



$$Y \approx \frac{77R + 150G + 29B}{2^8} = (77R + 150G + 29B) \ll 8$$



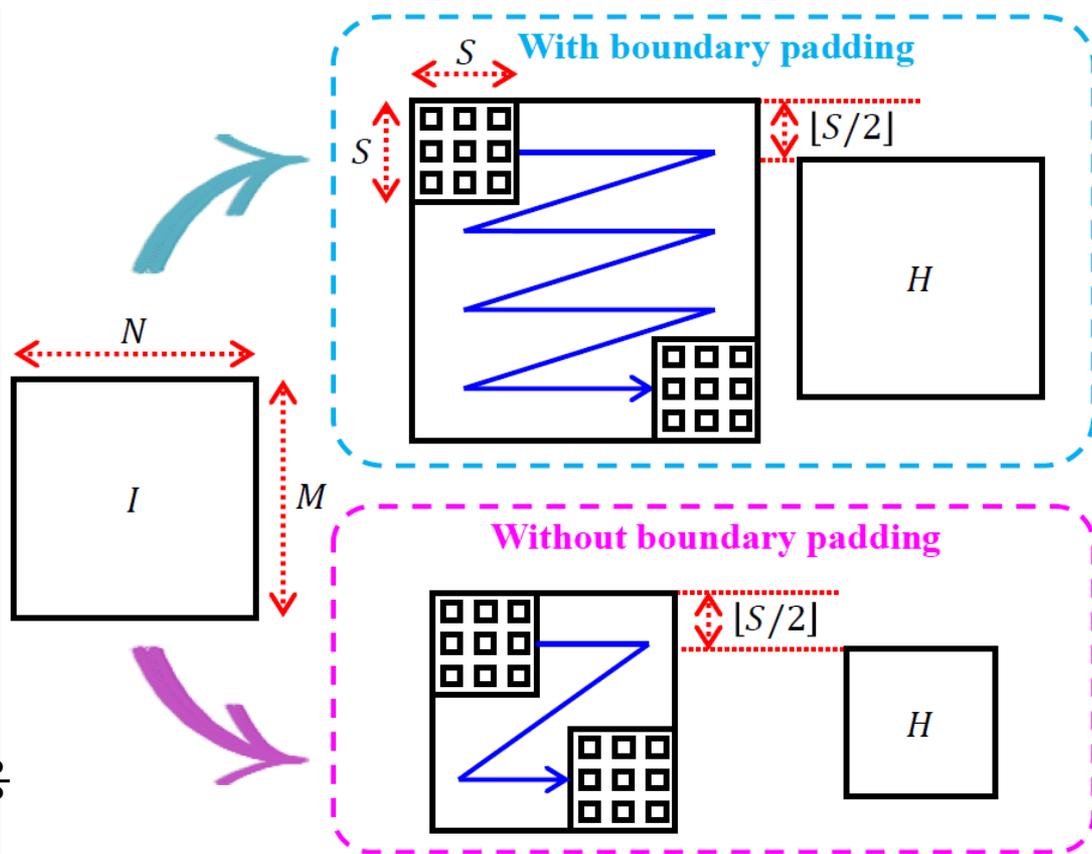
실습1: RGB to Gray



- 설계 방식
 - Python 이용하여 변환 공식 구현
 - Verilog 코드 작성
 - 테스트 데이터 생성 및 testbench 작성
 - ModelSim 이용하여 기능 시뮬레이션 수행
 - 집적 설계 가능성 확인(회로 합성 및 타이밍 시뮬레이션)
 - 하드웨어 IP 생성
 - 시스템 설계 및 FPGA 검증

실습2: 영상 필터링

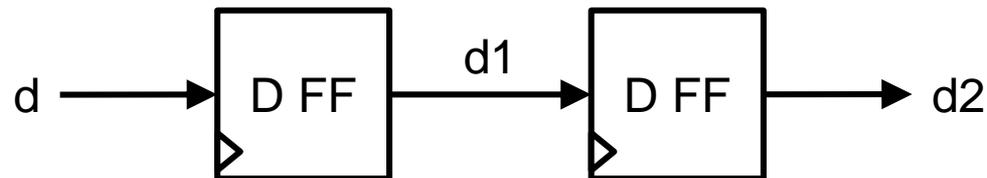
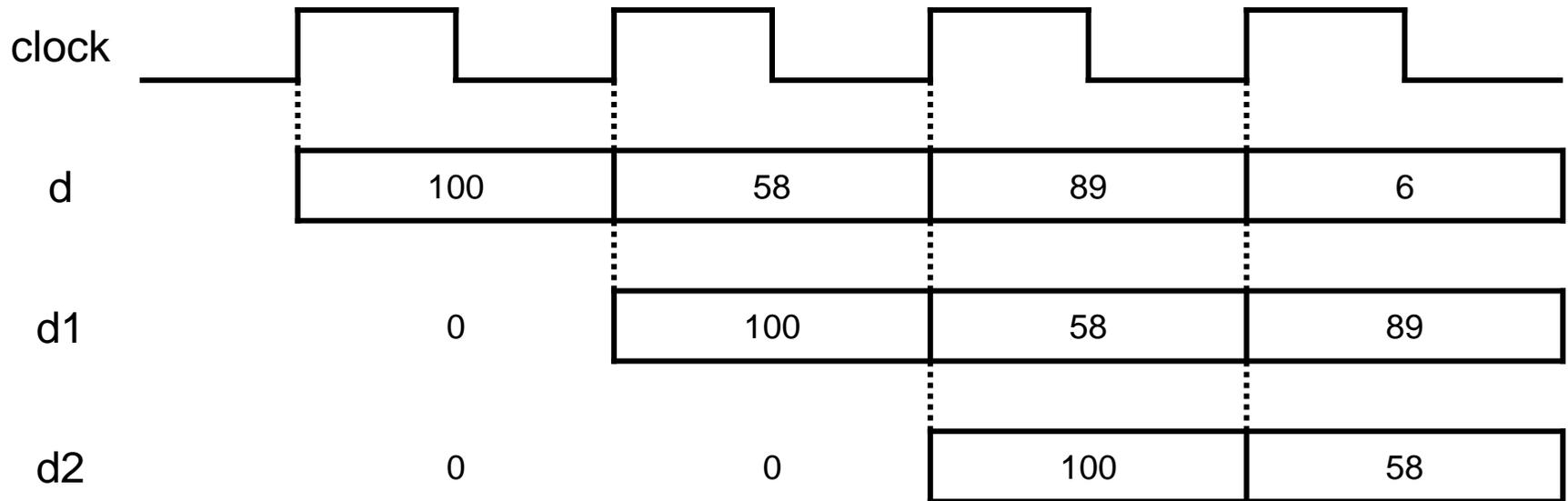
- 2D 필터
 - 라인 메모리 필요
- 1D 필터
 - 라인 메모리 불필요
- 실습 내용
 - 실습1에서 설계한 RGB to Gray 모듈 활용하여 영상의 gray 채널을 $\begin{bmatrix} 1 & 1 & 1 \\ 3 & 3 & 3 \end{bmatrix}$ 커널과 1D 필터 수행
 - Zero padding 방식 사용



실습2: 영상 필터링



- 1D 필터 구현 방식: **D FF 사용**





실습2: 영상 필터링

- 설계 방식
 - Python 이용하여 변환 공식 구현
 - Verilog 코드 작성
 - 테스트 데이터 생성 및 testbench 작성
 - ModelSim 이용하여 기능 시뮬레이션 수행
 - 집적 설계 가능성 확인(회로 합성 및 타이밍 시뮬레이션)
 - 하드웨어 IP 생성
 - 시스템 설계 및 FPGA 검증