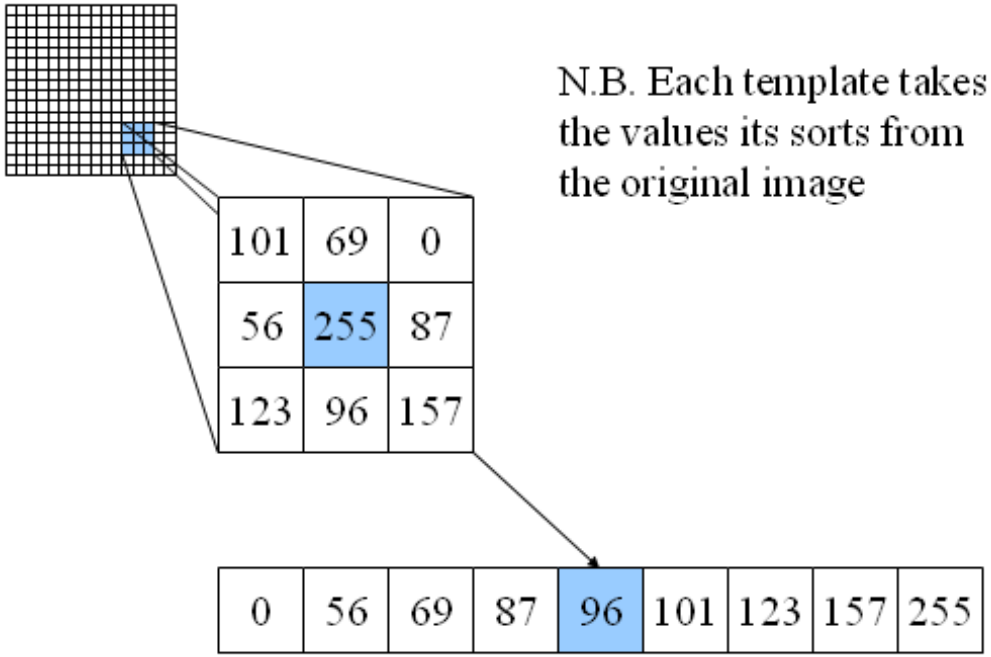
Final Project – Median Filter

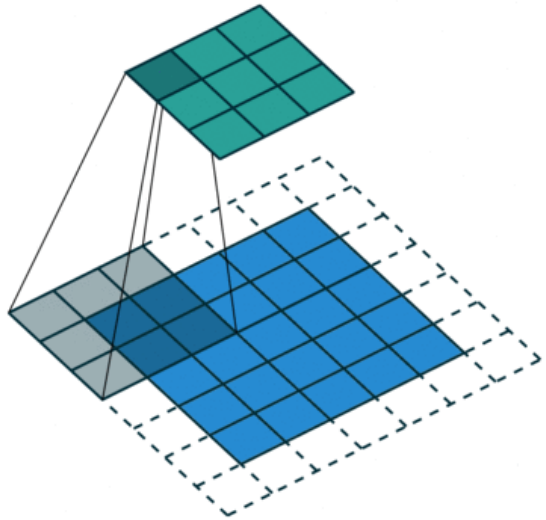
Median Filter

- Median Filter : '중앙값' 필터

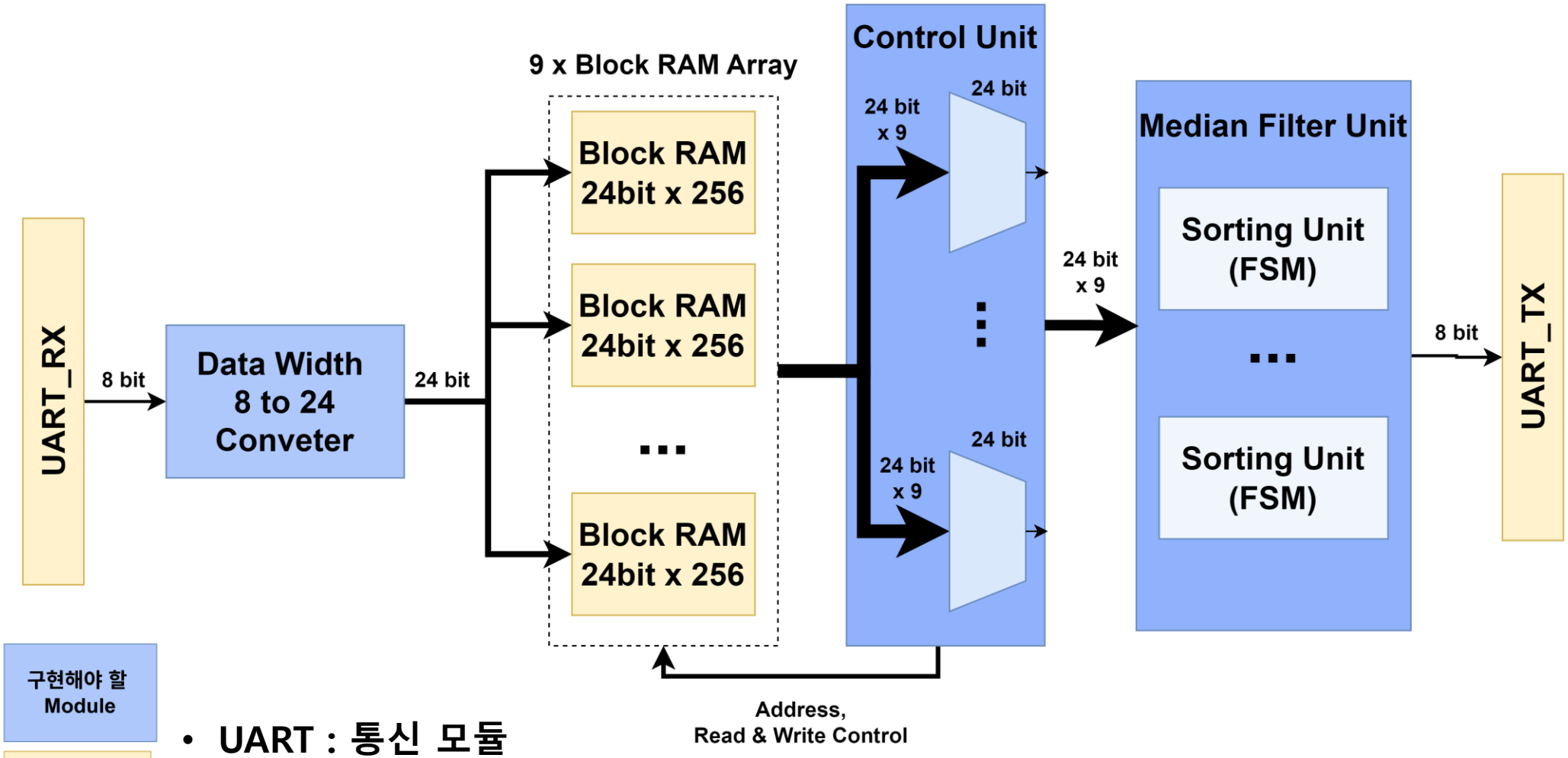
- 이미지에서 9칸의 pixel을 선택하고 값을 큰 순서대로 정렬
  →그 중 크기가 중앙인 값을 선택하여 저장
  → FSM을 활용하여 정렬 알고리즘 구현

- 테두리는 0으로 처리





구현해야 할 Architecture

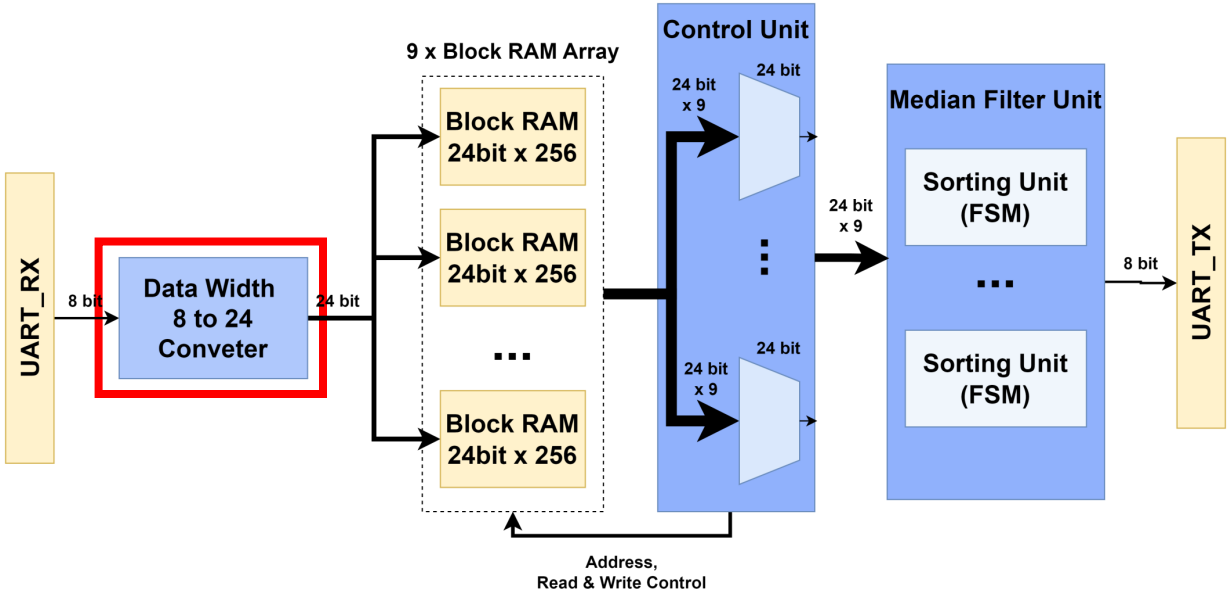


구현해야 할 Module

제공되는 Module

- UART : 통신 모듈
- BRAM : 저장 매체

Data Width Converter

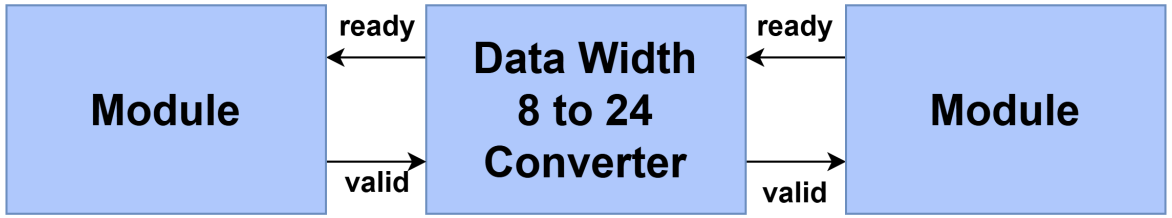


- 8Bit씩 데이터를 받아서 24bit 데이터 하나로 만든 후 Block RAM Array에 저장

  → **FSM으로 구현**

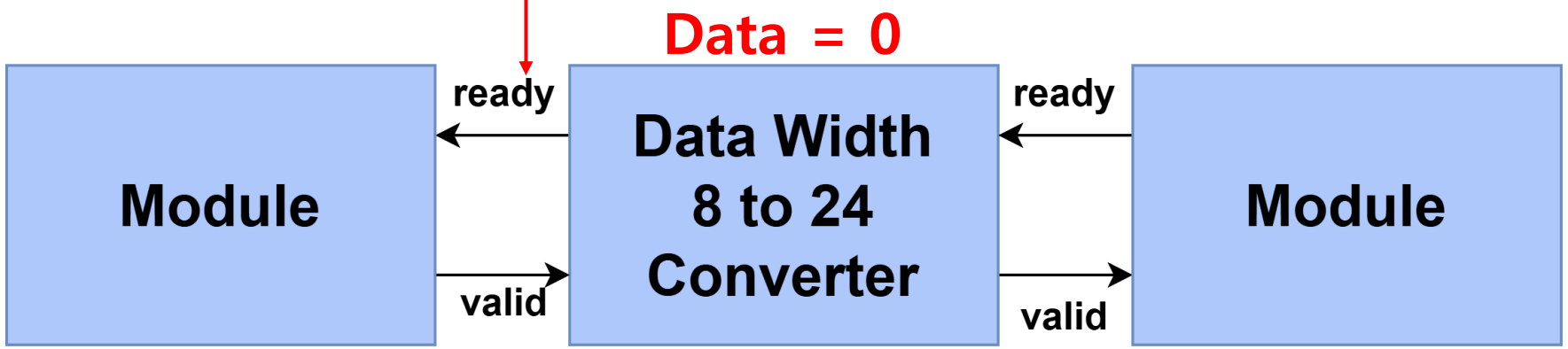
Valid & Ready Handshake

- 모듈간 통신 : 두 모듈이 준비되었을 때 데이터 전송
  - ready = 1



Valid & Ready Handshake

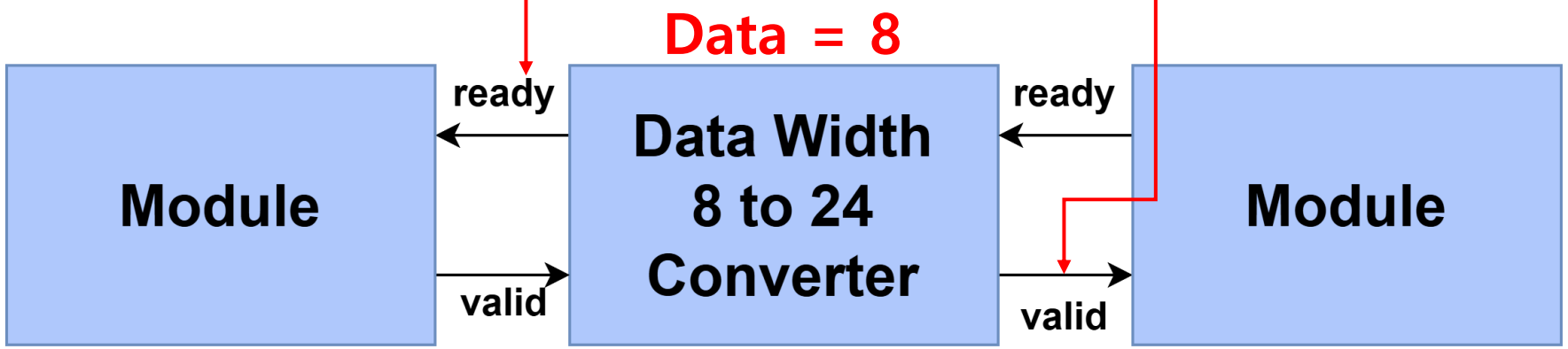
Converter 내부 Data = 0
→데이터 수신 가능
→ready = 1



Valid & Ready Handshake

Converter 내부 Data = 8
→데이터 수신 가능
→ready = 1

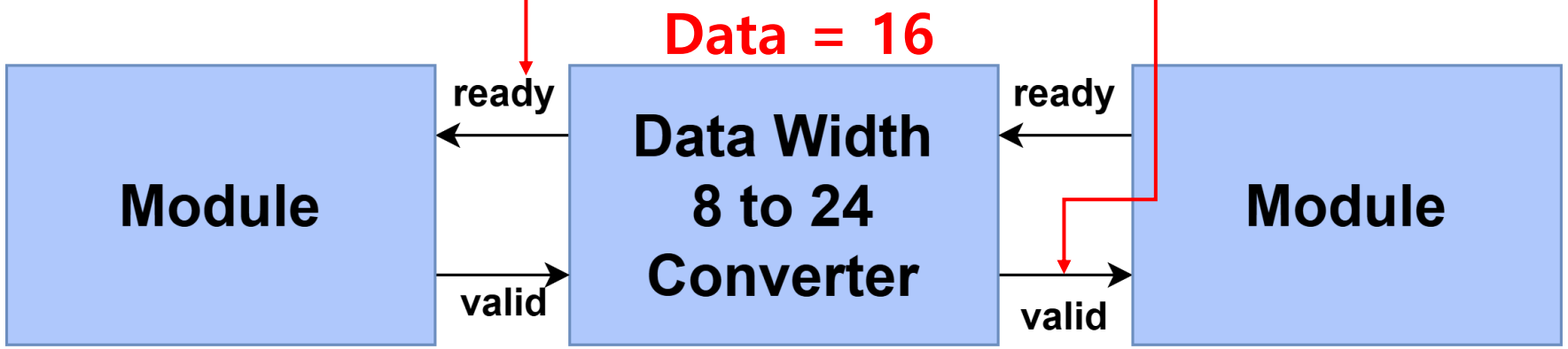
24bit Data 미완성
→valid = 0



Valid & Ready Handshake

Converter 내부 Data = 16
→데이터 수신 가능
→ready = 1

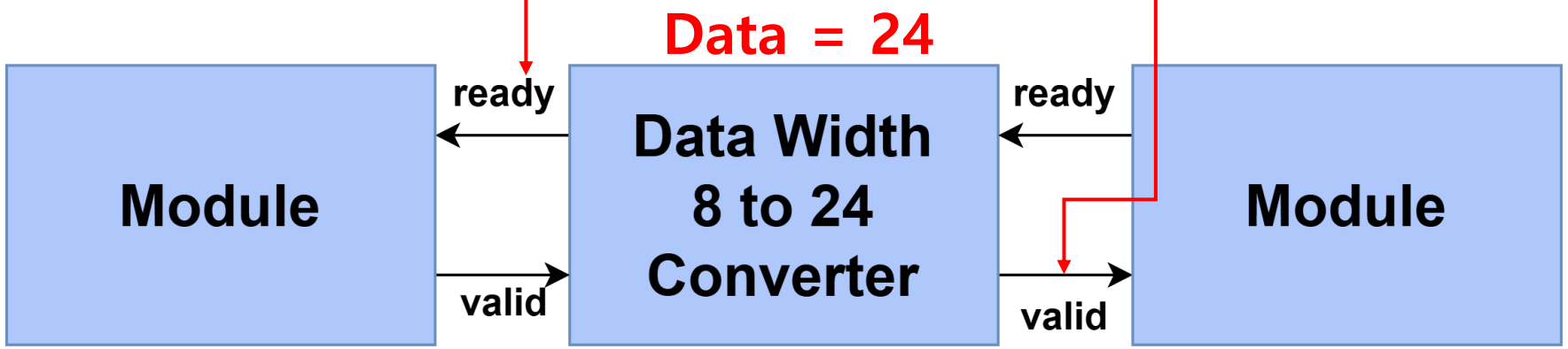
24bit Data 미완성
→valid = 0



Valid & Ready Handshake

Converter 내부 Data = 24
→데이터 수신 완료
→ready = 0,

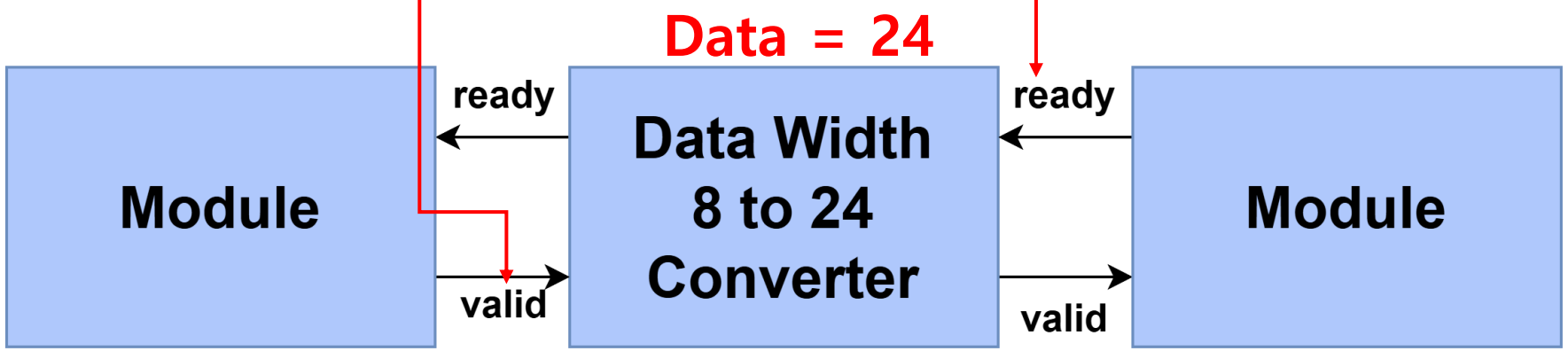
24bit Data 완성
→valid = 1



Valid & Ready Handshake

8bit * 3 전송 완료
→valid = 1

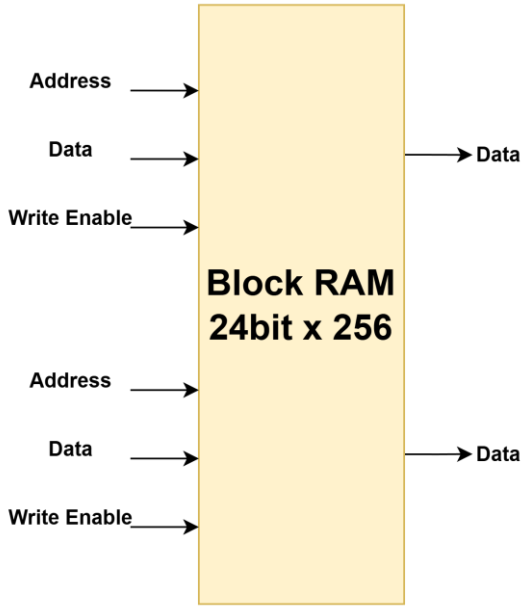
24bit Data 수신 및 Converter 수신 가능
→ready = 1



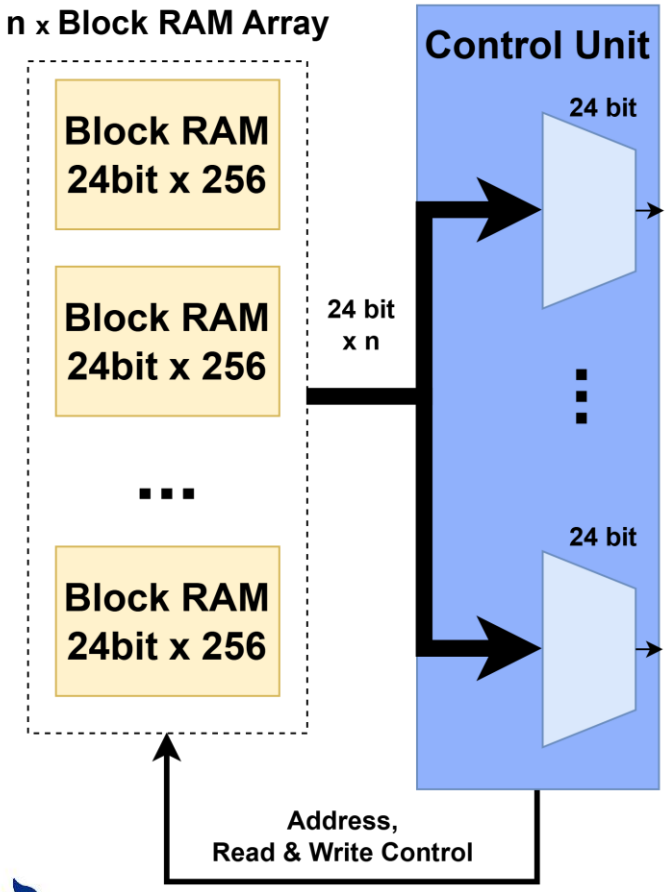
BRAM (Block RAM)

- BRAM의 용량은 36Kb (32b*1024)
- 이미지 전체를 저장하기엔 부족한 용량
- **적절한 개수를 사용하여 Median Filter 연산에 효율적으로 Data 전달**

```
module block_ram(
    input clk,

    input ena,
    input wea,
    input [7:0] addra,
    input [23:0] dina,
    output reg [23:0] douta,

    input enb,
    input web,
    input [7:0] addrb,
    input [23:0] dinb,
    output reg [23:0] doutb

);
```



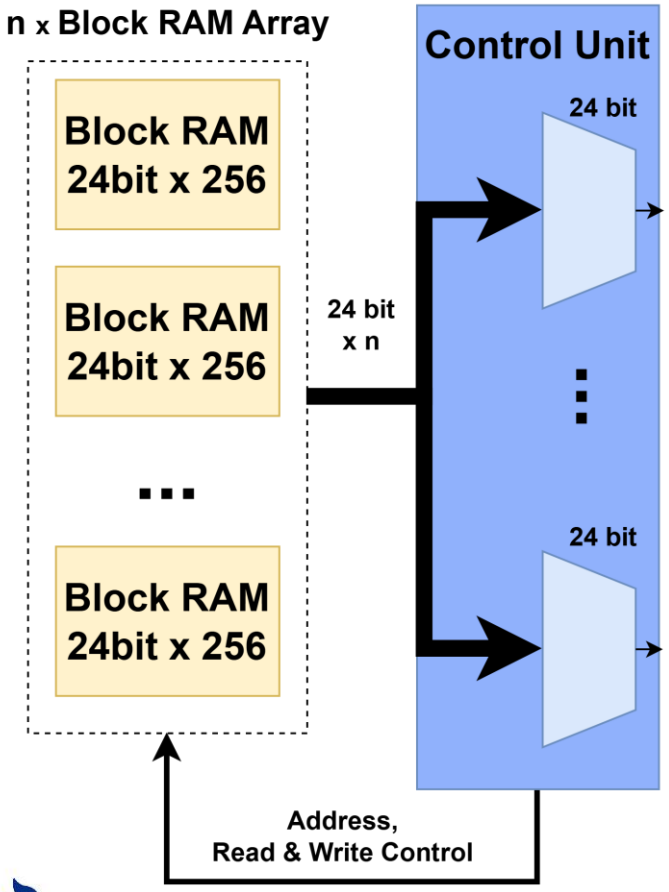
Control Unit



- 각각의 BRAM에 Read / Write 및 Address 제어
- 각 pixel을 기준으로 무엇이 들어가야 할지 결정

X	X	X	X	X	X
X	1	2	3	4	5
X	1	2	3	4	5
X	1	2	3	4	5
X	1	2	3	4	5
X	1	2	3	4	5
X	1	2	3	4	5

Control Unit




```
module control_unit_fsm (
    input clk,
    input resetn,

    input valid_in,
    output ready_in,

    input [24*9-1:0] mux_data_in,

    output reg valid_out,

    output [8*9-1:0] blkram_read_address,


    output reg [7:0] blkram_write_address,
    output reg [8:0] blkram_write_enable,

    output reg [24*9-1:0] mux_data_out
);


endmodule
```

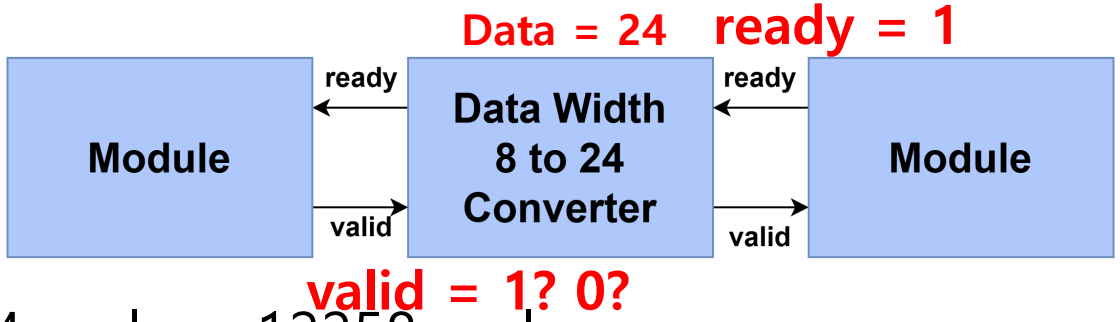
제약 사항

- 1. BRAM 사용 개수 제한
  - 9개로 사용 제한! 및 BRAM 수정 불가
- 2. Converter, Control Unit, Median Filter는 FSM으로 작성
  - 내부에서 호출하는 모듈의 경우 제약 없음
- 3. BRAM 외의 대용량 저장 방식 사용 금지

평가 기준

- **1. 전체 동작 여부**
  - **실패 시 모듈별 작동여부 확인으로 부분점수 부여**
  - **추가 testbench case들 (다른 이미지 및 랜덤 input valid 신호, reset으로 체크)**
- **2. 제약 사항 준수 확인**
  - BRAM 사용 제한 및 resource constraint 준수 여부 확인(utilization)
  - 제약 모듈에 대한 FSM으로 구현 했는지 확인대용량 레지스터 등의 편법 사용 여부
- **3. Latency**
  - 낭비되는 시간이 없는가?

    →최소로 구현 시 64*64*3+(64*1+2)+4 cycle = 12258 cycle



수업 진행 관련 안내

- 보드 관리를 위해 보드는 실습 시간에만 사용 가능
  - 11월 12일까지 Lab 진행
  - 11월 19일 ~ 12월 10일 파이널프로젝트 질의응답 / 보드 사용 가능 시간 (출석체크 X)
  - Tip) 한정된 보드 사용 시간 → 시뮬레이션까지 따로 짜고 실습 시간에 보드 사용 (수업 일정 참고하여 조별로 계획 잘 세우기!)
- 요구되는 결과물
  - 결과보고서 / 베릴로그 코드 (학번_이름.zip) (12월 17일까지 제출)
  - 작동되는 보드 (12월 17일 실습 시간에 현장 검사)

- 질문???