

# ARM 벡터 레지스터

정보컴퓨터공학과 권혁동

# Contents

ARM 프로세서

벡터 레지스터

결론



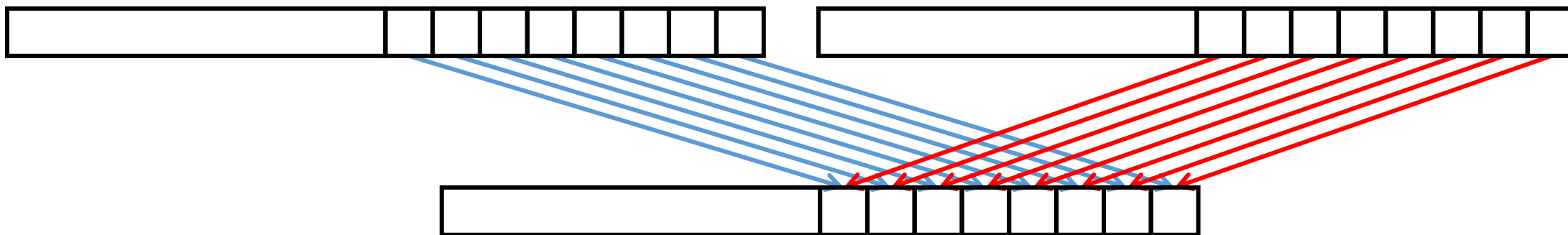
# ARM 프로세서

- ARM에서 제작하는 CPU 아키텍처
- Cortex-A 시리즈가 최신
- ARMv7: 32-bit
- **ARMv8: 64-bit (AArch64)**
- 두 종류의 레지스터로 분류
  - 스칼라 레지스터
  - **벡터 레지스터**

# 벡터 레지스터

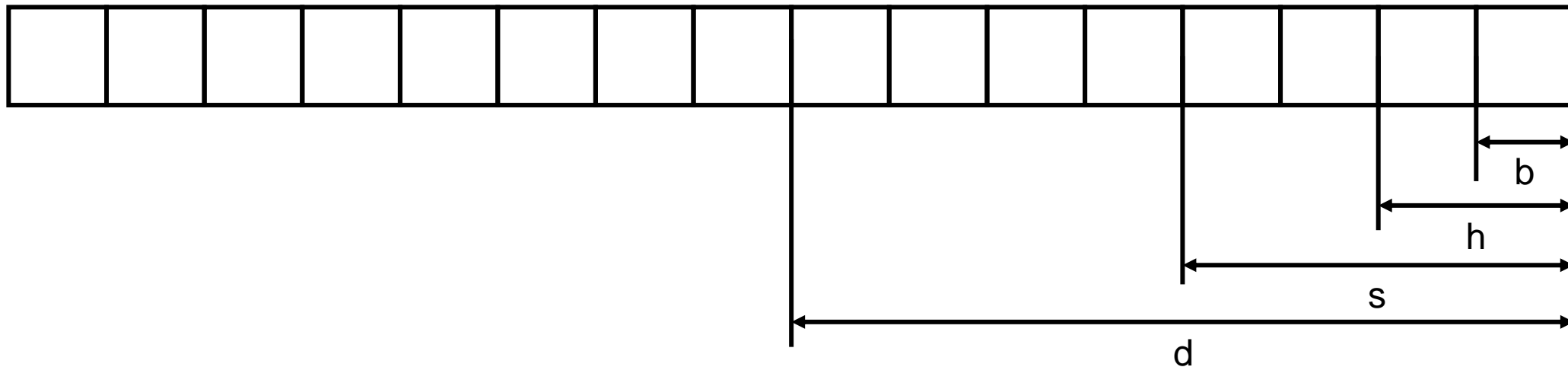
- 스칼라 레지스터는 일반적인 레지스터
  - 64-bit의 크기를 지니며, 31개 사용 가능
- **벡터 레지스터는 데이터의 병렬화가 가능한 레지스터**
  - 128-bit의 크기를 지니며, 32개 사용 가능
- 벡터 레지스터를 지원하는 벡터 인스트럭션 존재
  - 흔히 SIMD(Single Instruction Multiple Data)라 부름
- 스칼라 인스트럭션과 호환되지 않으므로 따로 사용

# 벡터 레지스터



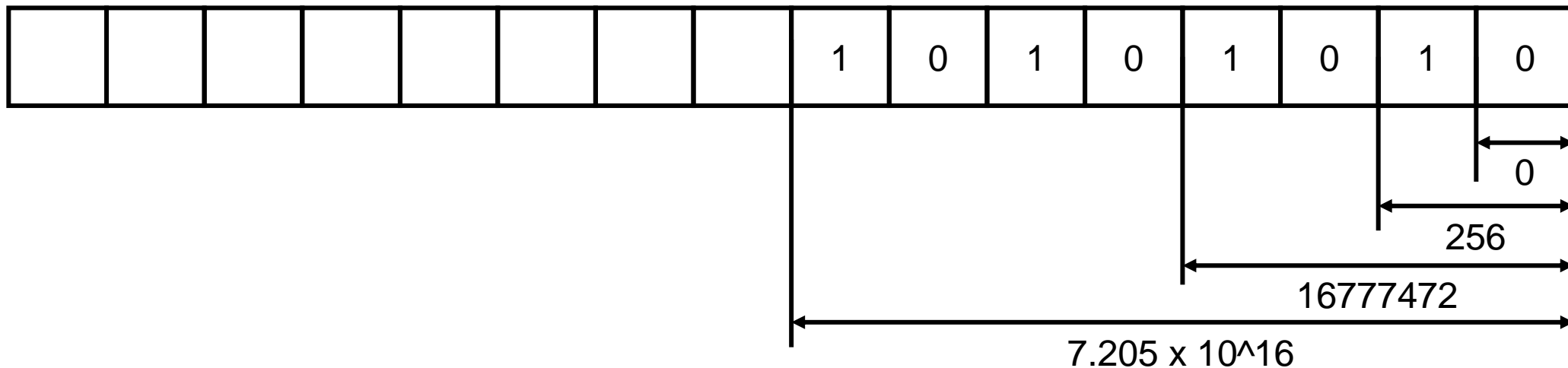
- 벡터 인스트럭션을 사용하여 병렬적으로 연산이 가능
  - 스칼라 레지스터 또는 AVR 같은 환경에서는 이와 같은 연산이 불가능

# 벡터 레지스터



- 벡터 레지스터는 v0-v31로 호칭됨
- 데이터 패킹을 어떻게 하는지에 따라 같은 연산자로 다른 값 도출
  - 패킹 단위: byte(b), half-word(h), single-word(s), double-word(d)

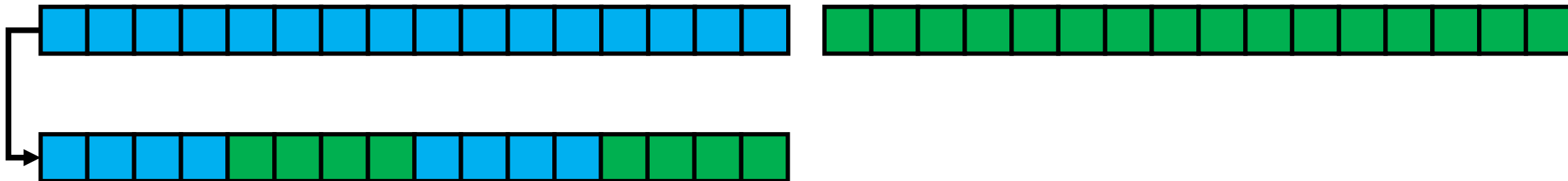
# 벡터 레지스터



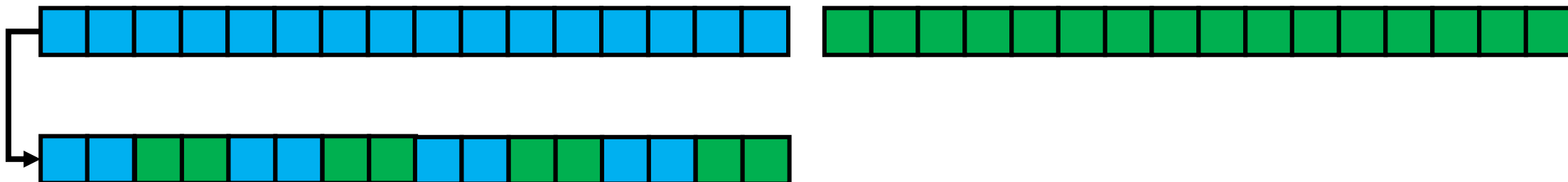
- **패킹 단위에 따라서** 같은 값이 저장되어 있어도,  
**다른 값으로 처리 가능**
- 따라서 같은 값이라도 패킹 단위에 따라 다른 결과 값을 도출 가능

# 벡터 레지스터

- UZP1.4s v0, v1, v2



- UZP1.8h v0, v1, v2





# 결론

- 벡터 레지스터의 구조 및 원리에 대해서 파악
- 벡터 레지스터로 병렬 구현이 가능한 이유를 확인
- 데이터 패킹 단위에 따라 각기 다른 결과를 도출 가능
- ARM 프로세서 상에서 최적 구현 시에 활용