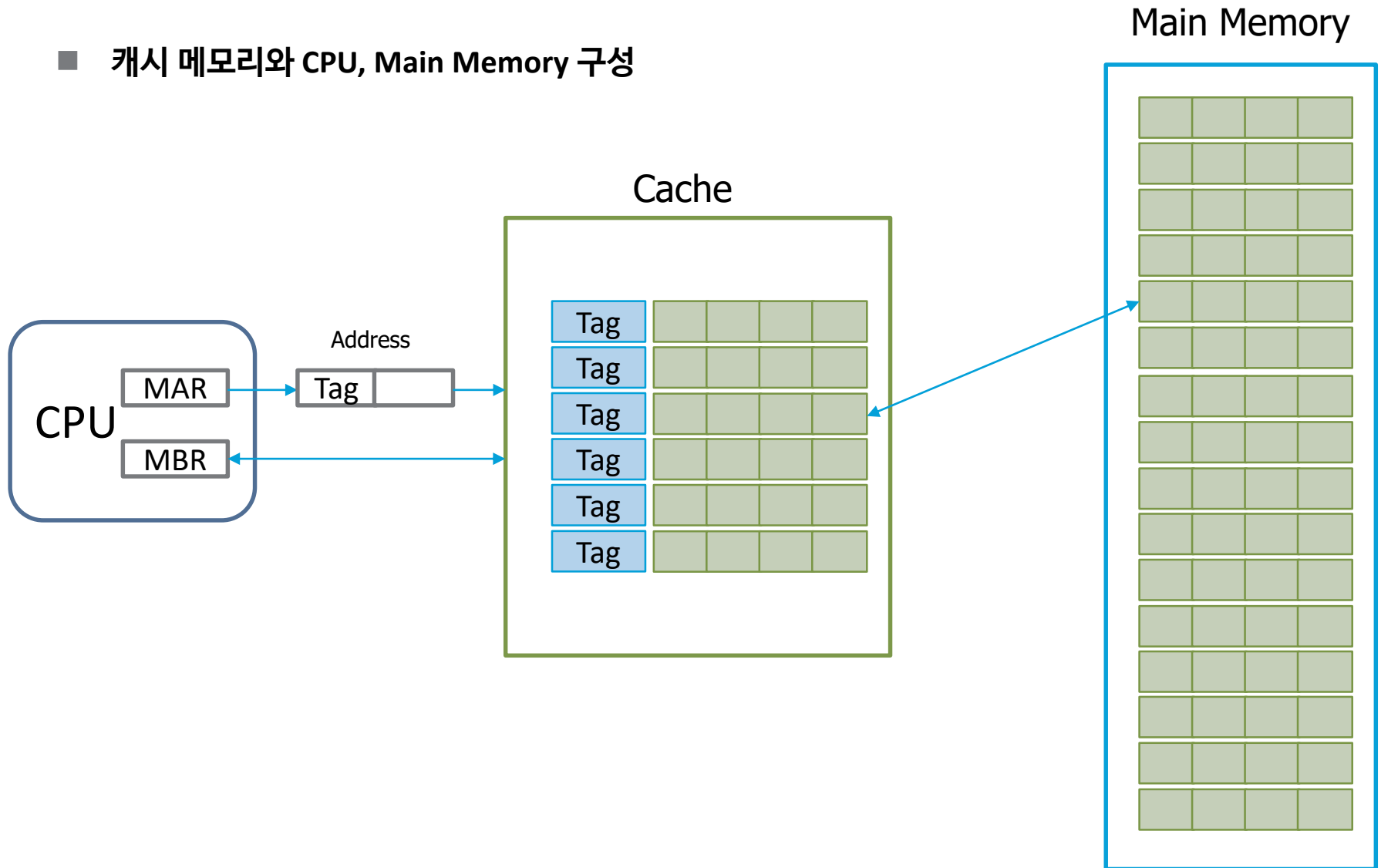


# 캐시 메모리의 태그(tag) 계산 방법

## ■ 캐시 메모리와 CPU, Main Memory 구성



# 캐시 메모리의 태그(tag) 계산 방법: Direct Mapping

- $\text{Tag} = (\text{Address} / w) / a = (\text{Address} / w) \gg m = \text{Address} \gg (\#bits_{\text{block}} + m)$ 
  - $\#bits_{\text{block}} = \log_2 w$
  - $w = \text{Block size}$
  - $m = \log_2 a$
  - $a = \text{Number of blocks in cache}$
- $\#bits_{\text{tag}} = \#bits_{\text{address}} - \#bits_{\text{block}} - m$
- Cache 적재시 Tag를 cache의 tag 필드에 저장
- $\text{Block index in memory} = \text{Address} / w = \text{Address} \gg \#bits_{\text{block}}$
- $\text{Block index in cache} = (\text{Address} / w) \% a$
- Hit 확인
  - `if (Address >> (#bitsblock + m) == Tag in Cache[Block index]) then Hit`  
`else Miss`

# 캐시 메모리의 태그(tag) 계산 방법: Fully-Associative Mapping

- $\text{Tag} = (\text{Address} / w) = \text{Address} \gg \#bits_{\text{block}}$ 
  - $w = \text{Block size}$
- $\#bits_{\text{tag}} = \#bits_{\text{address}} - \#bits_{\text{block}}$
- Cache 적재시 Tag를 cache의 tag 필드에 저장
- $\text{Block index in memory} = \text{Address} / w = \text{Address} \gg \#bits_{\text{block}}$
- Block index in cache: 정해져 있지 않음
- Hit 확인
  - if (Address  $\gg \#bits_{\text{block}} ==$  Cache의 모든 Tag중 같은 게 있으면) then Hit  
else Miss

# 캐시 메모리의 태그(tag) 계산 방법: k-way Set-associative Mapping

- **Tag = (Address / w) / S = (Address / L) >> h = Address >> (#bits<sub>block</sub> + h)**
  - w = Block size
  - $h = \log_2 S$
  - S = size of a set =  $2^k$
  - a = Number of blocks in cache
  - k = Number of sets
- **#bits<sub>tag</sub> = #bits<sub>address</sub> - #bits<sub>block</sub> - h**
- **Block index in memory = Address / L = Address >> #bits<sub>block</sub>**
- **Block index in cache = (Address / w) % S**
  - 집합 안에서는 block의 index
  - 어느 집합으로 들어갈지는 정해져 있지 않음
- **Hit 확인**
  - **if** (Address >> (#bits<sub>block</sub> + h) == Tag in Cache세트들[Block index in cache]) **then** Hit  
**else** Miss