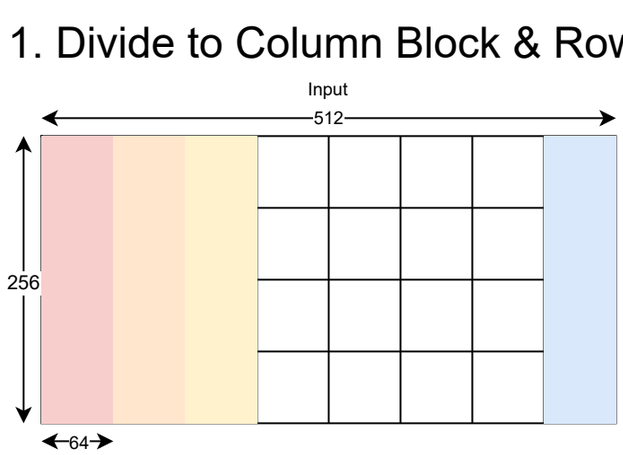
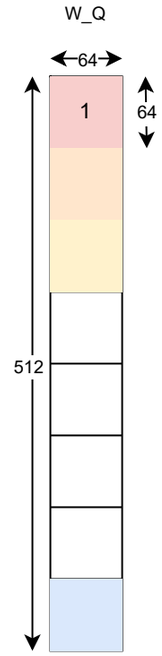
1. Divide to Column Block & Row Blcok



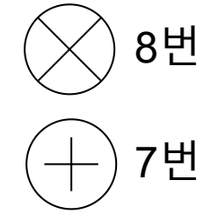
Divide to Column

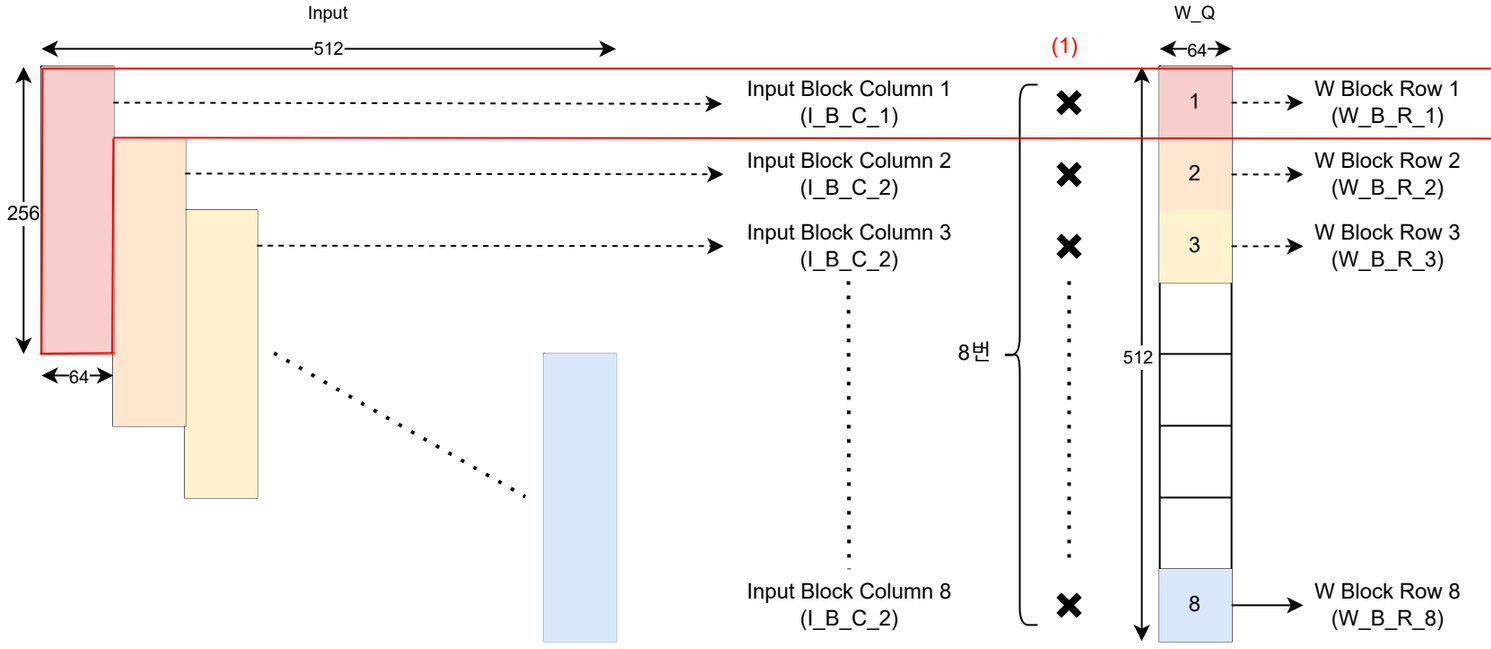
1 Column Block's Row = 64

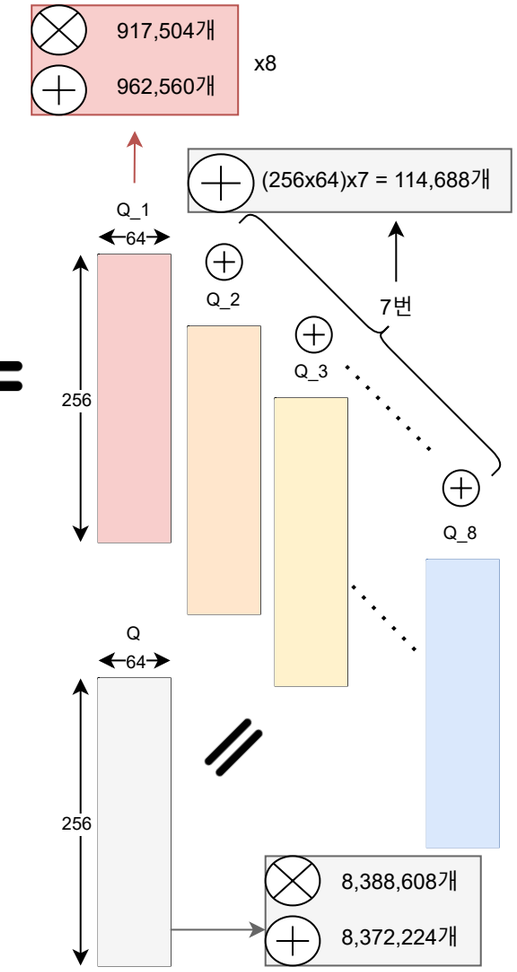


Divide to Row

1 Row Block's Column = 64

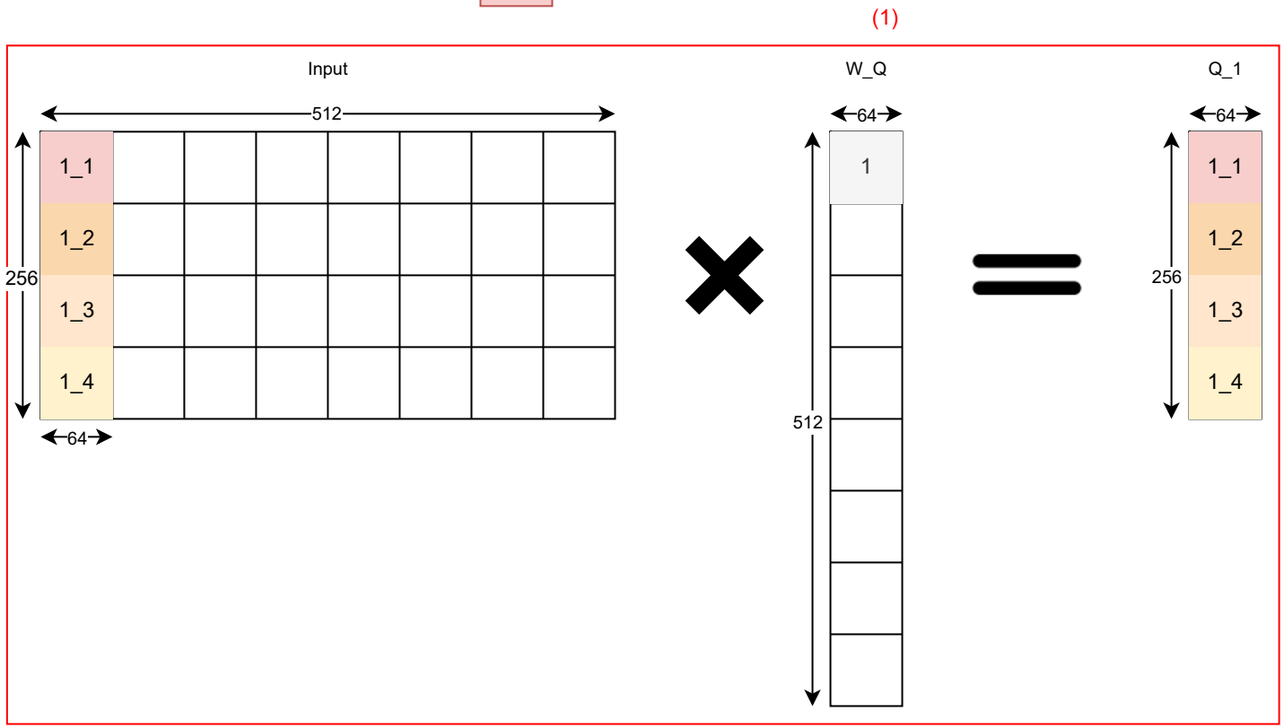


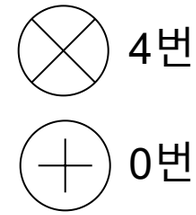


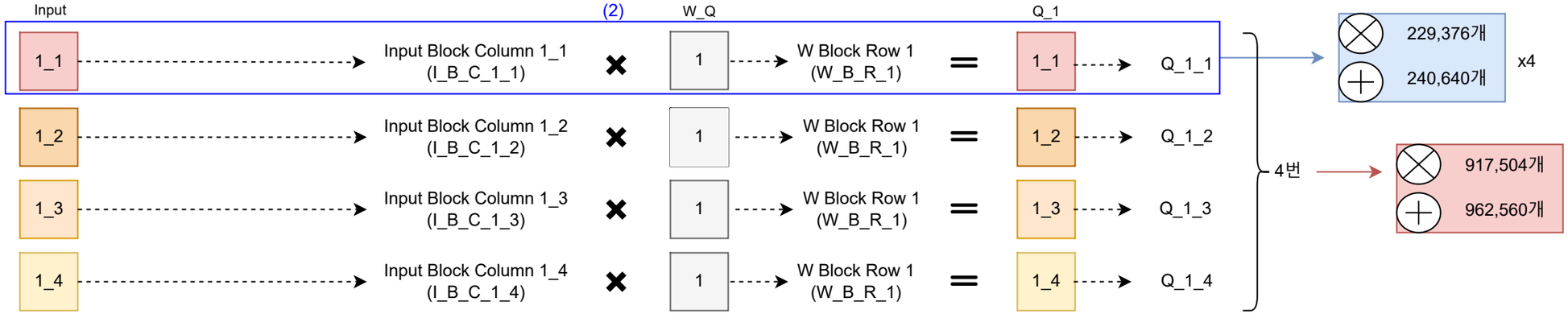


2. Divide to Block

Block(64x64)





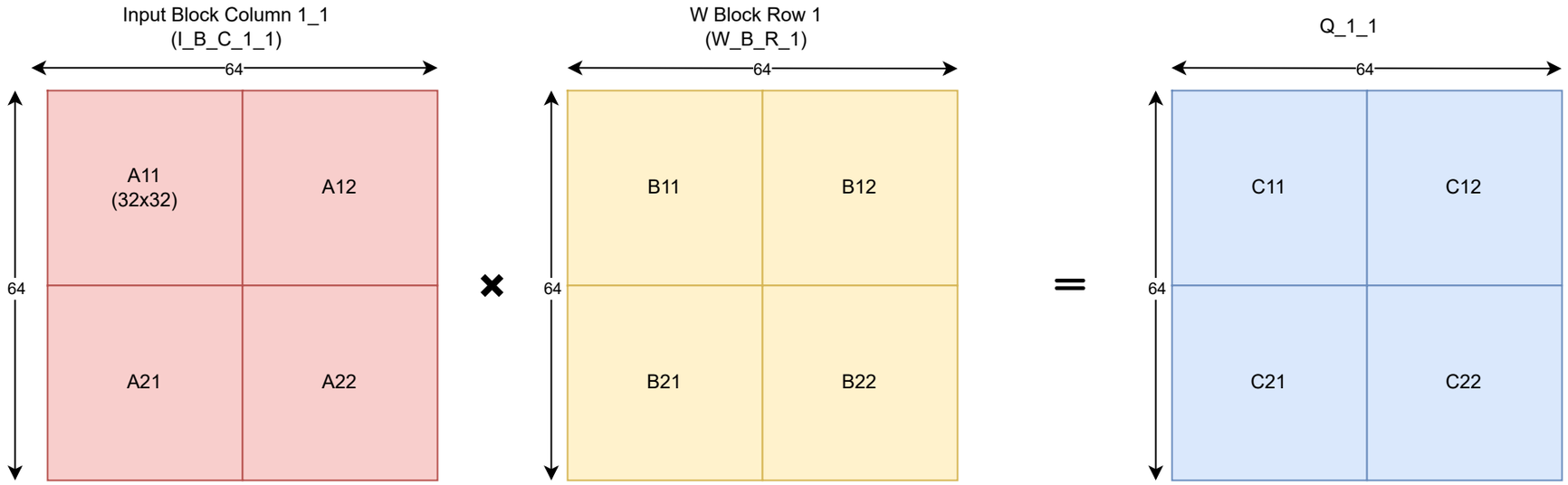


3. Multiplication Block Using Strassen Algorithm

n>=32 부터 시간차이 발생
64x64 행렬에서는 더 빠름

⊗ 229,376 번

⊕ 240,640번



Stage 1

VA1 = A11 + A22
VA2 = A21 + A22
VA3 = A11
VA4 = A22
VA5 = A11 + A12
VA6 = A21 - A11
VA7 = A12 - A22

덧셈: 5번

VB1 = B11 + B22
VB2 = B11
VB3 = BB12 - B22
VB4 = B21 - B11
VB5 = B22
VB6 = B11 + B12
VB7 = B21 + B22

덧셈: 5번

⊕ (nxn)x5x2 = (32x32)x5x2 = 10240개

Stage 2

V1 = VA1 * VB1
V2 = VA2 * VB2
V3 = VA3 * VB3
V4 = VA4 * VB4
V5 = VA5 * VB5
V6 = VA6 * VB6
V7 = VA7 * VB7

곱셈: 7번

⊗ (nxnxn)x7 = (32x32x32)x7 = 229,376개

⊕ {nx(n-1)xn}x7 = (32x31x32)x7 = 222,208개

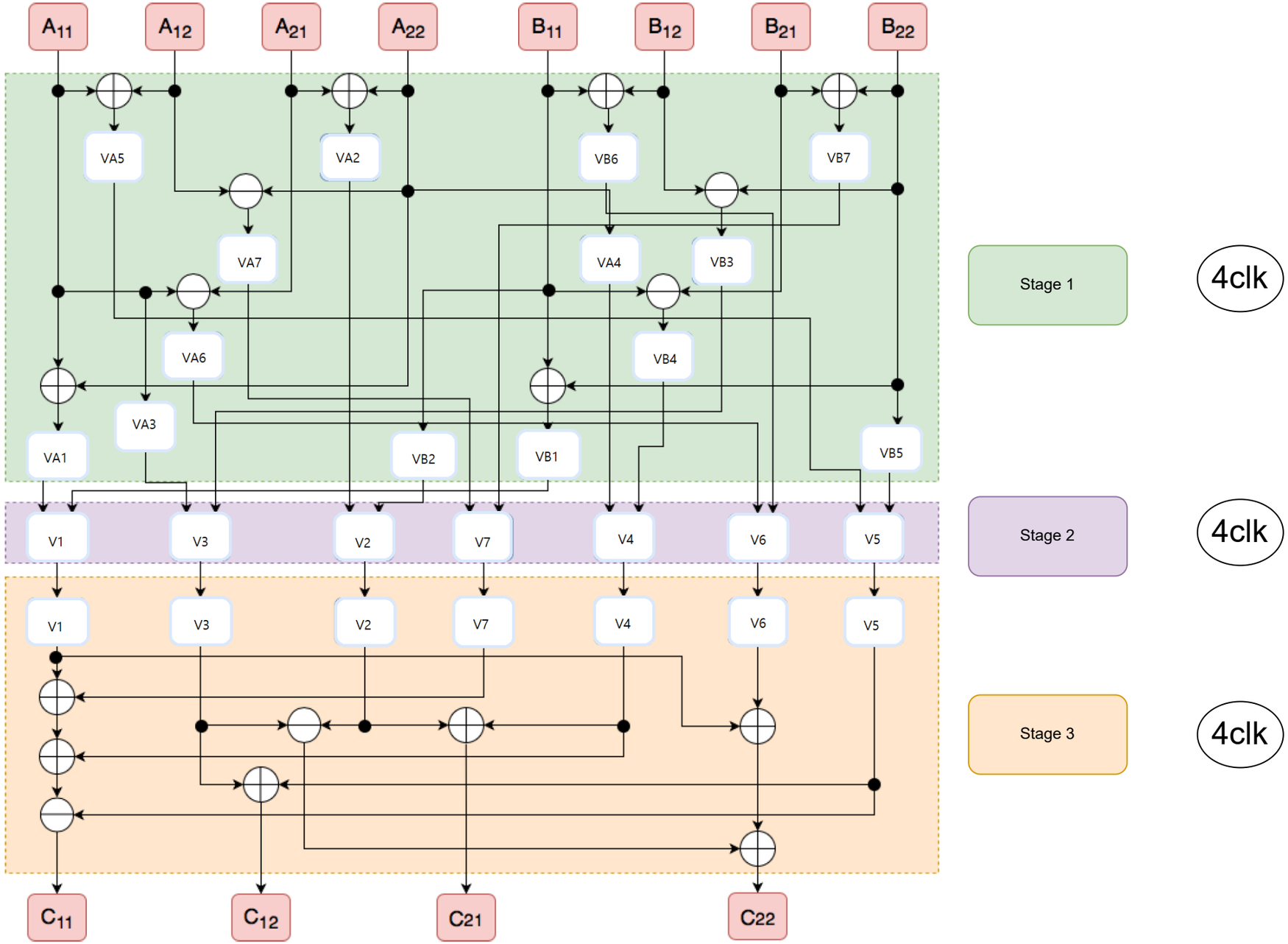
Stage 3

C11 = V1 + V4 -V5 +V7
C12 = V3 +V5
C21 = V2 + V4
C22 = V1 - V2 + V3 + V6

덧셈: 8번

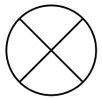
⊕ (nxn)x8 = (32x32)x8 = 8192개

3. Multiplication Block Using Strassen Algorithm



* Total Mul & Add

Naive Matrix Multiplier

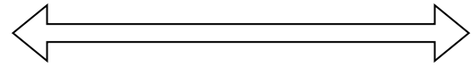


n x m x k = 256x512x64 = **8,388,608개**

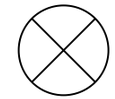
n x (m-1) x k = 256 x 511 x 64 = **8,372,224개**

장점: - H/W Size 작음

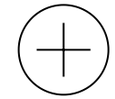
단점: - 연산기 개수 많음
- 클럭 수 많음
- 연산 속도 느림



Divide to Block using Strassen Algorithm



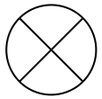
7,340,032개



7,815,168개

장점: - 연산기 개수 적음
- 클럭 수 적음
- 연산 속도 빠름

단점: - H/W size 큼



32bit floating point multiplier

32bit floating point adder