Overall	Block	Operation	Input Shape (H×W×C)	Output Shape (H×W×C) (XXS / XS / S)	Hardware Mapping
	Input	Image Input	-	256×256×3	DMA load → buffer
	Stem	Conv 3×3, stride=2	256×256×3	128×128×16	Conv kernel
	Stage 1	MV2 Block ×1	128×128×16	128×128×(16/32/32)	Uses MV2 master table
	Stage 2	MV2 Block, stride=2	128×128×(16 / 32 / 32)	64×64×(24 / 48 / 64)	Uses MV2 master table
	Stage 2	MV2 Block ×2	64×64×(24 / 48 / 64)	64×64×(24 / 48 / 64)	Uses MV2 master table
	Stage 3	MV2 Block, stride=2	64×64×(24 / 48 / 64)	32×32×(48 / 64 / 96)	Uses MV2 master table
	Stage 3	MobileViT Block (L=2)	32×32×(48 / 64 / 96)	32×32×(48 / 64 / 96) (d=64/96/144)	Uses MVB master table
	Stage 4	MV2 Block, stride=2	32×32×(48 / 64 / 96)	16×16×(64 / 80 / 128)	Uses MV2 master table
	Stage 4	MobileViT Block (L=4)	16×16×(64 / 80 / 128)	16×16×(64 / 80 / 128) (d=80/120/192)	Uses MVB master table
	Stage 5	MV2 Block, stride=2	16×16×(64 / 80 / 128)	8×8×(80 / 96 / 160)	Uses MV2 master table
	Stage 5	MobileViT Block (L=3)	8×8×(80 / 96 / 160)	8×8×(80 / 96 / 160) (d=96/144/240)	Uses MVB master table
	Head	Conv 1×1	8×8×(80 / 96 / 160)	8×8×(320 / 384 / 640)	Conv kernel
	Head	Global Pool	8×8×(320 / 384 / 640)	1×1×(320/384/640)	Pool unit
	Classifier	Linear	1×1×(320 / 384 / 640)	1×1×1000	Matmul kernel
	Chan	Outside	Innut > Outmut	Contract	Handware Manufac
	Step	Operation	Input → Output C	Output	Hardware Mapping Considerate
MV2	1 2	Expansion Conv 1×1 Batch Normalization	4C	4C (XXS: 2C) 4C	Conv kernel Batch Normalization Unit
	2	Nonlinearity	4C 4C	4C 4C	Activation unit (ReLU6 / Swish)
	4	Depthwise Conv 3×3	4C	4C (stride=1 or 2)	Conv kernel (channel-wise mode)
	5	Batch Normalization	4C	4C	Batch Normalization Unit
	6	Nonlinearity	4C	4C	Activation unit (ReLU6 / Swish)
	7	Projection Conv 1×1	4C	Cout	Conv kernel
	8	Batch Normalization	Cout	Cout	Batch Normalization Unit
	9	Nonlinearity	Cout → Cout	Cout	Activation unit (ReLU6 / Swish)
	10 (optional	<u> </u>	If stride=1 & same dims	Add unit	Add unit
	(,			
	Step	Operation	Input	Output	Hardware Mapping
	Step 1	Local Conv 3×3	H×W×C	H×W×C	Conv kernel
	Step 1 2	Local Conv 3×3 Pointwise Conv 1×1	H×W×C H×W×C	H×W×C H×W×d	Conv kernel Conv kernel
10/0	Step 1 2 3	Local Conv 3×3 Pointwise Conv 1×1 Unfold (Patchify)	H×W×C H×W×C H×W×d	H × W × C H × W × d Patches × d	Conv kernel Conv kernel DMA + buffer rearrange
MVB	Step 1 2 3 4	Local Conv 3×3 Pointwise Conv 1×1 Unfold (Patchify) Transformer Layer ×L	H × W × C H × W × C H × W × d Patches × d	H × W × C H × W × d Patches × d Patches × d	Conv kernel Conv kernel DMA + buffer rearrange See Transformer Table for hardware mapping
MVB	Step 1 2 3 4 5	Local Conv 3×3 Pointwise Conv 1×1 Unfold (Patchify) Transformer Layer ×L Fold (Reconstruct)	H × W × C H × W × C H × W × d Patches × d Patches × d	H × W × C H × W × d Patches × d Patches × d H × W × d	Conv kernel Conv kernel DMA + buffer rearrange See Transformer Table for hardware mapping DMA + buffer rearrange
MVB	Step 1 2 3 4 5 6	Local Conv 3×3 Pointwise Conv 1×1 Unfold (Patchify) Transformer Layer ×L Fold (Reconstruct) Pointwise Conv 1×1	H × W × C H × W × C H × W × d Patches × d Patches × d H × W × d	H × W × C H × W × d Patches × d Patches × d H × W × d H × W × C	Conv kernel Conv kernel DMA + buffer rearrange See Transformer Table for hardware mapping DMA + buffer rearrange Conv kernel
MVB	1 2 3 4 5 6 7	Local Conv 3×3 Pointwise Conv 1×1 Unfold (Patchify) Transformer Layer ×L Fold (Reconstruct) Pointwise Conv 1×1 Concatenation	H × W × C H × W × C H × W × d Patches × d Patches × d H × W × d H × W × C	H × W × C H × W × d Patches × d Patches × d H × W × d H × W × C H × W × C	Conv kernel Conv kernel DMA + buffer rearrange See Transformer Table for hardware mapping DMA + buffer rearrange Conv kernel Concat unit
MVB	Step 1 2 3 4 5 6 7 8	Local Conv 3×3 Pointwise Conv 1×1 Unfold (Patchify) Transformer Layer ×L Fold (Reconstruct) Pointwise Conv 1×1	H × W × C H × W × C H × W × d Patches × d Patches × d H × W × d	H × W × C H × W × d Patches × d Patches × d H × W × d H × W × C	Conv kernel Conv kernel DMA + buffer rearrange See Transformer Table for hardware mapping DMA + buffer rearrange Conv kernel
MVB	1 2 3 4 5 6 7	Local Conv 3×3 Pointwise Conv 1×1 Unfold (Patchify) Transformer Layer ×L Fold (Reconstruct) Pointwise Conv 1×1 Concatenation	H × W × C H × W × C H × W × d Patches × d Patches × d H × W × d H × W × C H × W × 2C	H × W × C H × W × d Patches × d Patches × d H × W × d H × W × C H × W × C H × W × C	Conv kernel Conv kernel DMA + buffer rearrange See Transformer Table for hardware mapping DMA + buffer rearrange Conv kernel Concat unit Conv kernal
MVB	1 2 3 4 5 6 7 8	Local Conv 3×3 Pointwise Conv 1×1 Unfold (Patchify) Transformer Layer ×L Fold (Reconstruct) Pointwise Conv 1×1 Concatenation Conv 3x3	H × W × C H × W × C H × W × d Patches × d Patches × d H × W × d H × W × C	H × W × C H × W × d Patches × d Patches × d H × W × d H × W × C H × W × C	Conv kernel Conv kernel DMA + buffer rearrange See Transformer Table for hardware mapping DMA + buffer rearrange Conv kernel Concat unit
MVB	1 2 3 4 5 6 7 8	Local Conv 3×3 Pointwise Conv 1×1 Unfold (Patchify) Transformer Layer ×L Fold (Reconstruct) Pointwise Conv 1×1 Concatenation Conv 3x3 Operation LayerNorm (LN1)	H × W × C H × W × C H × W × d Patches × d Patches × d H × W × d H × W × C H × W × C Input Shape	H × W × C H × W × d Patches × d Patches × d H × W × d H × W × C H × W × C Output Shape Tokens × d	Conv kernel Conv kernel DMA + buffer rearrange See Transformer Table for hardware mapping DMA + buffer rearrange Conv kernel Concat unit Conv kernal Hardware Mapping / Accelerator View Element-wise ops: mean/variance reduction (adder tree + divider) + scale/shift (mult/add)
MVB	1 2 3 4 5 6 7 8 Step 1 2	Local Conv 3×3 Pointwise Conv 1×1 Unfold (Patchify) Transformer Layer ×L Fold (Reconstruct) Pointwise Conv 1×1 Concatenation Conv 3x3 Operation LayerNorm (LN1) Linear (Q projection)	H × W × C H × W × C H × W × d Patches × d Patches × d H × W × C H × W × C H × W × C Tokens × d Tokens × d	H × W × C H × W × d Patches × d Patches × d H × W × d H × W × C H × W × C Output Shape Tokens × d Tokens × d	Conv kernel Conv kernel DMA + buffer rearrange See Transformer Table for hardware mapping DMA + buffer rearrange Conv kernel Concat unit Conv kernal Hardware Mapping / Accelerator View Element-wise ops: mean/variance reduction (adder tree + divider) + scale/shift (mult/add) Matmul kernel (weights = [d × d _k])
MVB	1 2 3 4 5 6 7 8	Local Conv 3×3 Pointwise Conv 1×1 Unfold (Patchify) Transformer Layer ×L Fold (Reconstruct) Pointwise Conv 1×1 Concatenation Conv 3x3 Operation LayerNorm (LN1) Linear (Q projection) Linear (K projection)	H × W × C H × W × C H × W × d Patches × d Patches × d H × W × d H × W × C H × W × C Tokens × d Tokens × d Tokens × d	H × W × C H × W × d Patches × d Patches × d H × W × d H × W × C H × W × C Output Shape Tokens × d Tokens × d _k	Conv kernel Conv kernel DMA + buffer rearrange See Transformer Table for hardware mapping DMA + buffer rearrange Conv kernel Concat unit Conv kernal Hardware Mapping / Accelerator View Element-wise ops: mean/variance reduction (adder tree + divider) + scale/shift (mult/add) Matmul kernel (weights = [d × d _k])
MVB	1 2 3 4 5 6 7 8 Step 1 2	Local Conv 3×3 Pointwise Conv 1×1 Unfold (Patchify) Transformer Layer ×L Fold (Reconstruct) Pointwise Conv 1×1 Concatenation Conv 3x3 Operation LayerNorm (LN1) Linear (Q projection) Linear (K projection) Linear (V projection)	H × W × C H × W × C H × W × d Patches × d Patches × d H × W × C H × W × C H × W × C Tokens × d Tokens × d Tokens × d Tokens × d	H × W × C H × W × d Patches × d Patches × d H × W × d H × W × C H × W × C Output Shape Tokens × d Tokens × d _k Tokens × d _v	Conv kernel Conv kernel DMA + buffer rearrange See Transformer Table for hardware mapping DMA + buffer rearrange Conv kernel Concat unit Conv kernal Hardware Mapping / Accelerator View Element-wise ops: mean/variance reduction (adder tree + divider) + scale/shift (mult/add) Matmul kernel (weights = [d × d _k]) Matmul kernel Matmul kernel
MVB	1 2 3 4 5 6 7 8 Step 1 2	Local Conv 3×3 Pointwise Conv 1×1 Unfold (Patchify) Transformer Layer ×L Fold (Reconstruct) Pointwise Conv 1×1 Concatenation Conv 3x3 Operation LayerNorm (LN1) Linear (Q projection) Linear (K projection) Linear (V projection) Q × K ^T (Attention scores)	H × W × C H × W × C H × W × d Patches × d Patches × d H × W × d H × W × C H × W × 2C Input Shape Tokens × d Tokens × d Tokens × d Tokens × d	H × W × C H × W × d Patches × d Patches × d H × W × d H × W × C H × W × C Output Shape Tokens × d	Conv kernel Conv kernel DMA + buffer rearrange See Transformer Table for hardware mapping DMA + buffer rearrange Conv kernel Concat unit Conv kernal Hardware Mapping / Accelerator View Element-wise ops: mean/variance reduction (adder tree + divider) + scale/shift (mult/add) Matmul kernel (weights = [d × d _k]) Matmul kernel Matmul kernel Matmul kernel Matmul kernel (systolic array), heavy compute O(L ² · d _k)
MVB	1 2 3 4 5 6 7 8 Step 1 2	Local Conv 3×3 Pointwise Conv 1×1 Unfold (Patchify) Transformer Layer ×L Fold (Reconstruct) Pointwise Conv 1×1 Concatenation Conv 3x3 Operation LayerNorm (LN1) Linear (Q projection) Linear (V projection) Linear (V projection) Q × K ^T (Attention scores) Softmax	H × W × C H × W × C H × W × d Patches × d Patches × d H × W × d H × W × C H × W × 2C Input Shape Tokens × d	H × W × C H × W × d Patches × d Patches × d H × W × d H × W × C H × W × C Output Shape Tokens × d Tokens × d _k Tokens × d _v Tokens × Tokens Tokens × Tokens	Conv kernel Conv kernel DMA + buffer rearrange See Transformer Table for hardware mapping DMA + buffer rearrange Conv kernel Concat unit Conv kernal Hardware Mapping / Accelerator View Element-wise ops: mean/variance reduction (adder tree + divider) + scale/shift (mult/add) Matmul kernel (weights = [d × d _k]) Matmul kernel Matmul kernel Matmul kernel Matmul kernel (systolic array), heavy compute O(L ² · d _k) Exp LUT/polynomial + row reduction + reciprocal multiply
	1 2 3 4 5 6 7 8 Step 1 2	Local Conv 3×3 Pointwise Conv 1×1 Unfold (Patchify) Transformer Layer ×L Fold (Reconstruct) Pointwise Conv 1×1 Concatenation Conv 3x3 Operation LayerNorm (LN1) Linear (Q projection) Linear (K projection) Linear (V projection) Q × K ^T (Attention scores)	H × W × C H × W × C H × W × d Patches × d Patches × d H × W × d H × W × C H × W × 2C Input Shape Tokens × d Tokens × d Tokens × d Tokens × d	H × W × C H × W × d Patches × d Patches × d H × W × d H × W × C H × W × C Output Shape Tokens × d	Conv kernel Conv kernel DMA + buffer rearrange See Transformer Table for hardware mapping DMA + buffer rearrange Conv kernel Concat unit Conv kernal Hardware Mapping / Accelerator View Element-wise ops: mean/variance reduction (adder tree + divider) + scale/shift (mult/add) Matmul kernel (weights = [d × d _k]) Matmul kernel Matmul kernel Matmul kernel Matmul kernel (systolic array), heavy compute O(L ² · d _k)
	1 2 3 4 5 6 7 8 Step 1 2	Local Conv 3×3 Pointwise Conv 1×1 Unfold (Patchify) Transformer Layer ×L Fold (Reconstruct) Pointwise Conv 1×1 Concatenation Conv 3x3 Operation LayerNorm (LN1) Linear (Q projection) Linear (K projection) Linear (V projection) Q × K ^T (Attention scores) Softmax A × V (apply attention) Linear (output proj)	H × W × C H × W × C H × W × d Patches × d Patches × d H × W × d H × W × C H × W × C Tokens × d Tokens × Tokens Tokens × Tokens Tokens × Tokens Tokens × Tokens	H × W × C H × W × d Patches × d Patches × d H × W × d H × W × C H × W × C Output Shape Tokens × d Tokens × d _k Tokens × d _k Tokens × Tokens Tokens × Tokens Tokens × d _v Tokens × d _v	Conv kernel Conv kernel DMA + buffer rearrange See Transformer Table for hardware mapping DMA + buffer rearrange Conv kernel Concat unit Conv kernal Hardware Mapping / Accelerator View Element-wise ops: mean/variance reduction (adder tree + divider) + scale/shift (mult/add) Matmul kernel (weights = [d × d _k]) Matmul kernel Matmul kernel Matmul kernel Matmul kernel (systolic array), heavy compute O(L ² · d _k) Exp LUT/polynomial + row reduction + reciprocal multiply Matmul kernel Matmul kernel Matmul kernel
	1 2 3 4 5 6 7 8 8 9	Local Conv 3×3 Pointwise Conv 1×1 Unfold (Patchify) Transformer Layer ×L Fold (Reconstruct) Pointwise Conv 1×1 Concatenation Conv 3x3 Operation LayerNorm (LN1) Linear (Q projection) Linear (K projection) Linear (V projection) Q × K ^T (Attention scores) Softmax A × V (apply attention) Linear (output proj) Residual Add (skip)	H × W × C H × W × C H × W × d Patches × d Patches × d H × W × d H × W × C H × W × C Tokens × d Tokens × Tokens Tokens × Tokens Tokens × Tokens Tokens × Tokens	H × W × C H × W × d Patches × d Patches × d H × W × d H × W × C H × W × C Output Shape Tokens × d Tokens × d _k Tokens × d _k Tokens × Tokens Tokens × Tokens Tokens × d _v Tokens × d _v Tokens × d Tokens × Tokens	Conv kernel Conv kernel DMA + buffer rearrange See Transformer Table for hardware mapping DMA + buffer rearrange Conv kernel Concat unit Conv kernal Hardware Mapping / Accelerator View Element-wise ops: mean/variance reduction (adder tree + divider) + scale/shift (mult/add) Matmul kernel (weights = [d × d _k]) Matmul kernel Matmul kernel Matmul kernel (systolic array), heavy compute O(L ² · d _k) Exp LUT/polynomial + row reduction + reciprocal multiply Matmul kernel Matmul kernel Elementwise add unit
	1 2 3 4 5 6 7 8 8 9 10	Local Conv 3×3 Pointwise Conv 1×1 Unfold (Patchify) Transformer Layer ×L Fold (Reconstruct) Pointwise Conv 1×1 Concatenation Conv 3x3 Operation LayerNorm (LN1) Linear (Q projection) Linear (V projection) Linear (V projection) Q × K ^T (Attention scores) Softmax A × V (apply attention) Linear (output proj) Residual Add (skip) LayerNorm (LN2)	H × W × C H × W × C H × W × d Patches × d Patches × d H × W × d H × W × C H × W × C H × W × 2C Input Shape Tokens × d Tokens × Tokens	H × W × C H × W × d Patches × d Patches × d H × W × d H × W × C H × W × C Output Shape Tokens × d Tokens × d _k Tokens × d _k Tokens × Tokens Tokens × Tokens Tokens × Tokens Tokens × d Tokens × d Tokens × Tokens	Conv kernel Conv kernel DMA + buffer rearrange See Transformer Table for hardware mapping DMA + buffer rearrange Conv kernel Concat unit Conv kernal Hardware Mapping / Accelerator View Element-wise ops: mean/variance reduction (adder tree + divider) + scale/shift (mult/add) Matmul kernel (weights = [d × d _k]) Matmul kernel Exp LUT/polynomial + row reduction + reciprocal multiply Matmul kernel Matmul kernel Elementwise add unit Same as LN1
	1 2 3 4 5 6 7 8 8 9 10 11	Local Conv 3×3 Pointwise Conv 1×1 Unfold (Patchify) Transformer Layer ×L Fold (Reconstruct) Pointwise Conv 1×1 Concatenation Conv 3x3 Operation LayerNorm (LN1) Linear (Q projection) Linear (V projection) Linear (V projection) Q × K ^T (Attention scores) Softmax A × V (apply attention) Linear (output proj) Residual Add (skip) LayerNorm (LN2) Linear (FFN expansion)	H × W × C H × W × C H × W × d Patches × d Patches × d H × W × d H × W × C H × W × C H × W × 2C Input Shape Tokens × d Tokens × Tokens	H × W × C H × W × d Patches × d Patches × d H × W × d H × W × C H × W × C Output Shape Tokens × d Tokens × d _k Tokens × Tokens Tokens × Tokens Tokens × d _v Tokens × d Tokens × d Tokens × Tokens	Conv kernel Conv kernel DMA + buffer rearrange See Transformer Table for hardware mapping DMA + buffer rearrange Conv kernel Concat unit Conv kernal Hardware Mapping / Accelerator View Element-wise ops: mean/variance reduction (adder tree + divider) + scale/shift (mult/add) Matmul kernel (weights = [d × d _k]) Matmul kernel Matmul kernel Matmul kernel (systolic array), heavy compute O(L ² · d _k) Exp LUT/polynomial + row reduction + reciprocal multiply Matmul kernel Matmul kernel Elementwise add unit Same as LN1 Matmul kernel (larger width)
	1 2 3 4 5 6 7 8 8 9 10 11 12	Local Conv 3×3 Pointwise Conv 1×1 Unfold (Patchify) Transformer Layer ×L Fold (Reconstruct) Pointwise Conv 1×1 Concatenation Conv 3x3 Operation LayerNorm (LN1) Linear (Q projection) Linear (V projection) Linear (V projection) Q × K ^T (Attention scores) Softmax A × V (apply attention) Linear (output proj) Residual Add (skip) LayerNorm (LN2) Linear (FFN expansion) Activation (GELU/ReLU)	H × W × C H × W × C H × W × d Patches × d Patches × d H × W × d H × W × C H × W × C H × W × 2C Input Shape Tokens × d Tokens × Tokens Tokens × d Tokens × d Tokens × d	H × W × C H × W × d Patches × d Patches × d H × W × d H × W × C H × W × C H × W × C Output Shape Tokens × d Tokens × d Tokens × d _k Tokens × Tokens Tokens × Tokens Tokens × d Tokens × d Tokens × d Tokens × Tokens	Conv kernel Conv kernel DMA + buffer rearrange See Transformer Table for hardware mapping DMA + buffer rearrange Conv kernel Concat unit Conv kernal Hardware Mapping / Accelerator View Element-wise ops: mean/variance reduction (adder tree + divider) + scale/shift (mult/add) Matmul kernel (weights = [d × d _k]) Matmul kernel Matmul kernel Matmul kernel (systolic array), heavy compute O(L ² · d _k) Exp LUT/polynomial + row reduction + reciprocal multiply Matmul kernel Matmul kernel Elementwise add unit Same as LN1 Matmul kernel (larger width) Nonlinear unit (approx poly/LUT)
	1 2 3 4 5 6 7 8 8 9 10 11	Local Conv 3×3 Pointwise Conv 1×1 Unfold (Patchify) Transformer Layer ×L Fold (Reconstruct) Pointwise Conv 1×1 Concatenation Conv 3x3 Operation LayerNorm (LN1) Linear (Q projection) Linear (V projection) Linear (V projection) Q × K ^T (Attention scores) Softmax A × V (apply attention) Linear (output proj) Residual Add (skip) LayerNorm (LN2) Linear (FFN expansion)	H × W × C H × W × C H × W × d Patches × d Patches × d H × W × d H × W × C H × W × C H × W × 2C Input Shape Tokens × d Tokens × Tokens	H × W × C H × W × d Patches × d Patches × d H × W × d H × W × C H × W × C Output Shape Tokens × d Tokens × d _k Tokens × Tokens Tokens × Tokens Tokens × d _v Tokens × d Tokens × d Tokens × Tokens	Conv kernel Conv kernel DMA + buffer rearrange See Transformer Table for hardware mapping DMA + buffer rearrange Conv kernel Concat unit Conv kernal Hardware Mapping / Accelerator View Element-wise ops: mean/variance reduction (adder tree + divider) + scale/shift (mult/add) Matmul kernel (weights = [d × d _k]) Matmul kernel Matmul kernel Matmul kernel (systolic array), heavy compute O(L ² · d _k) Exp LUT/polynomial + row reduction + reciprocal multiply Matmul kernel Matmul kernel Elementwise add unit Same as LN1 Matmul kernel (larger width)