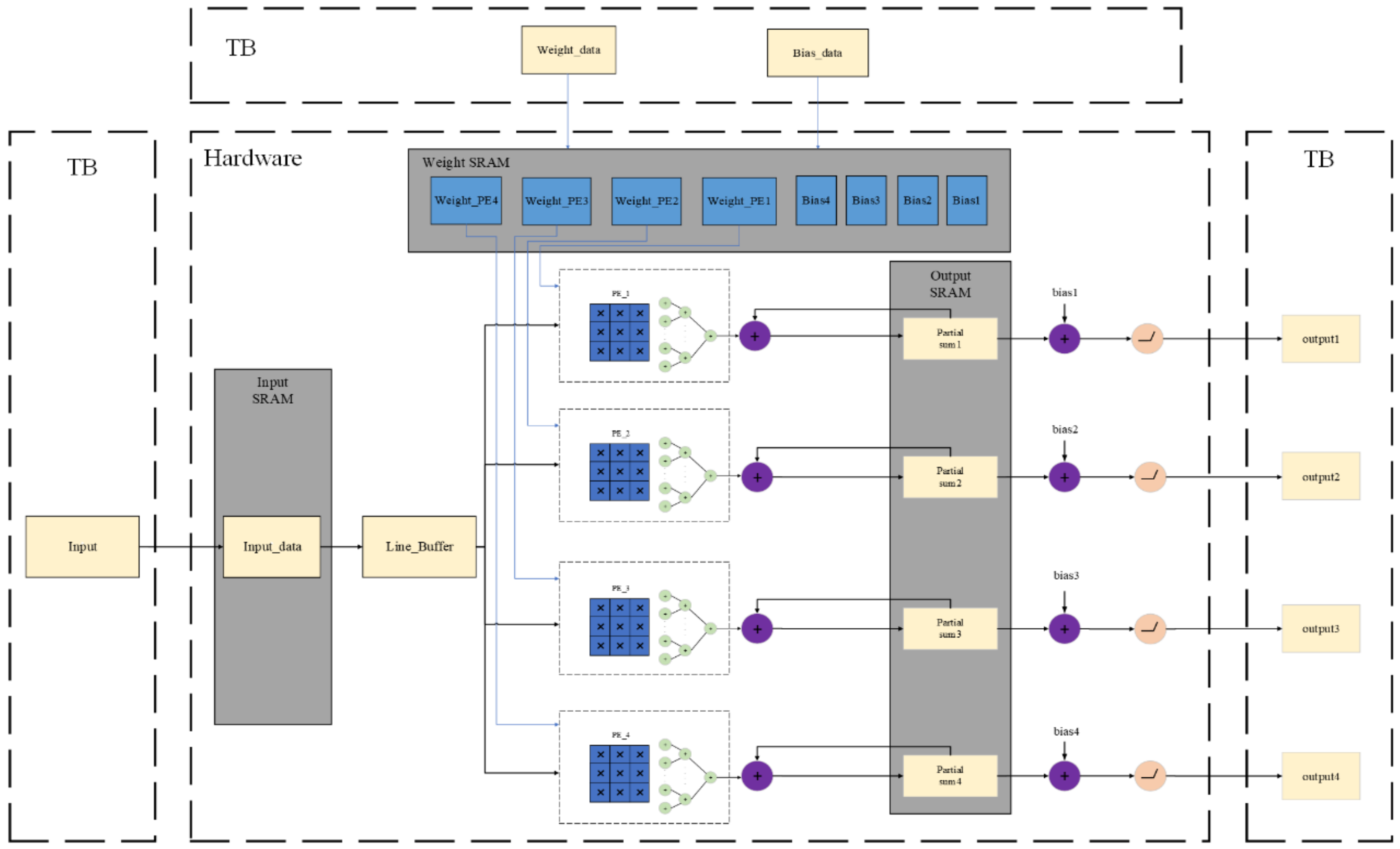
HDL HW 6: Memory Generation for the VGG CNN Accelerator

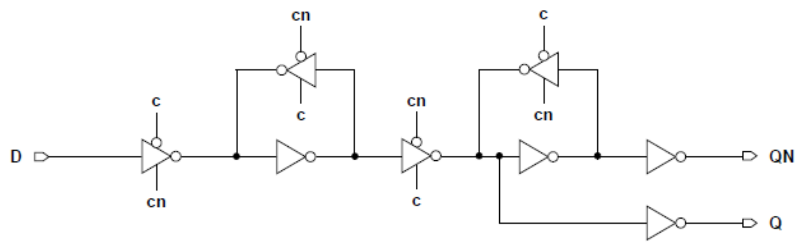
Outlines

- almost same as previous homework except
  - use memory compiler to generate hardware input/output/weight SRAM



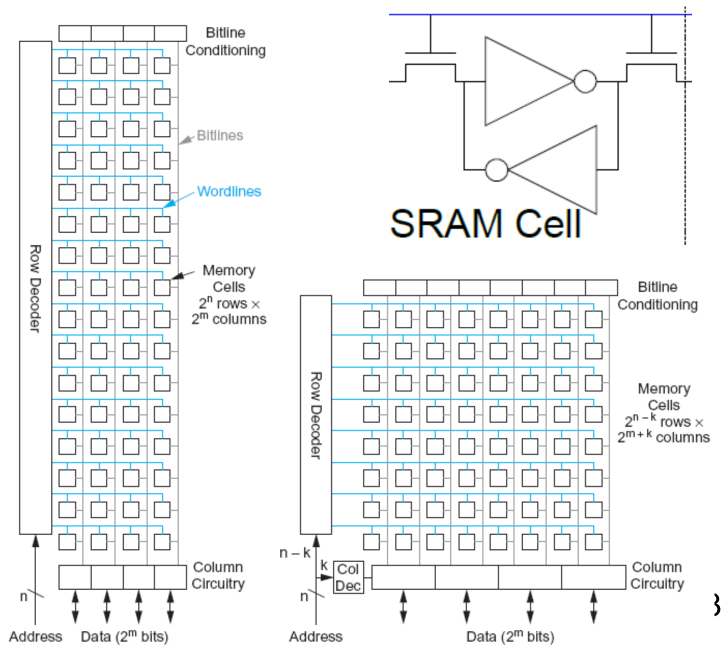
RAM vs. Registers

- Registers
  - use array to model registers
  - e.g., reg [31:0] R [0:15];
    - ✓ register file with 16 registers, each with 32 bits
    - ✓ realized using 16*32 FFs
  - many register locations can be accessed simultaneously
    - ✓ e.g., R[3] = R[0] + R]1] + R[2];
  - much more area than RAM
- RAM
  - need memory compiler tool to generate RAM
  - only the specified address can be accessed at a time
  - much smaller area than registers



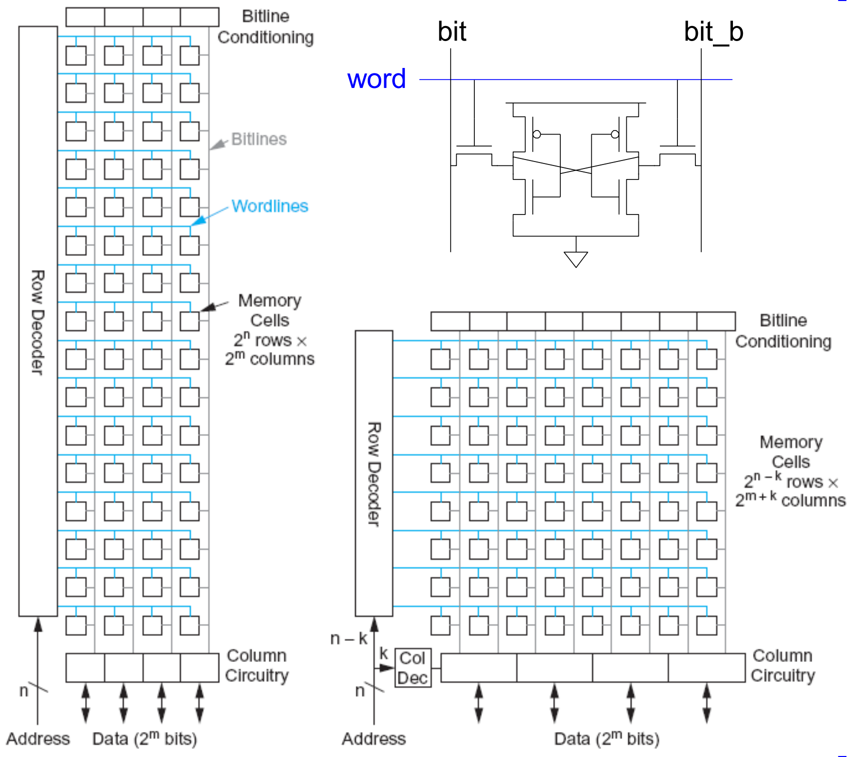
24 transistors in a flip-flop

6 transistors in a sram cell



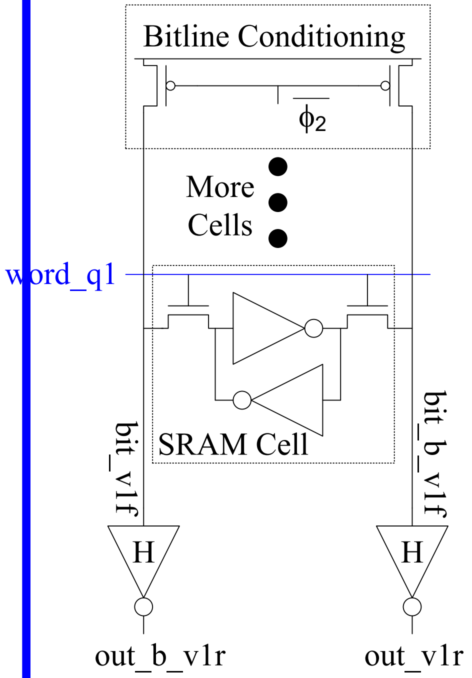
Memory Array Architecture

- 2^n *words* of 2^m *bits* each
- If n >> m, fold by 2^k into fewer *rows* of more *columns*
  - need both row decoder and column decoder
- Good regularity – easy for automatic generation
  - e.g., memory compiler
- Very high density if good cells are used

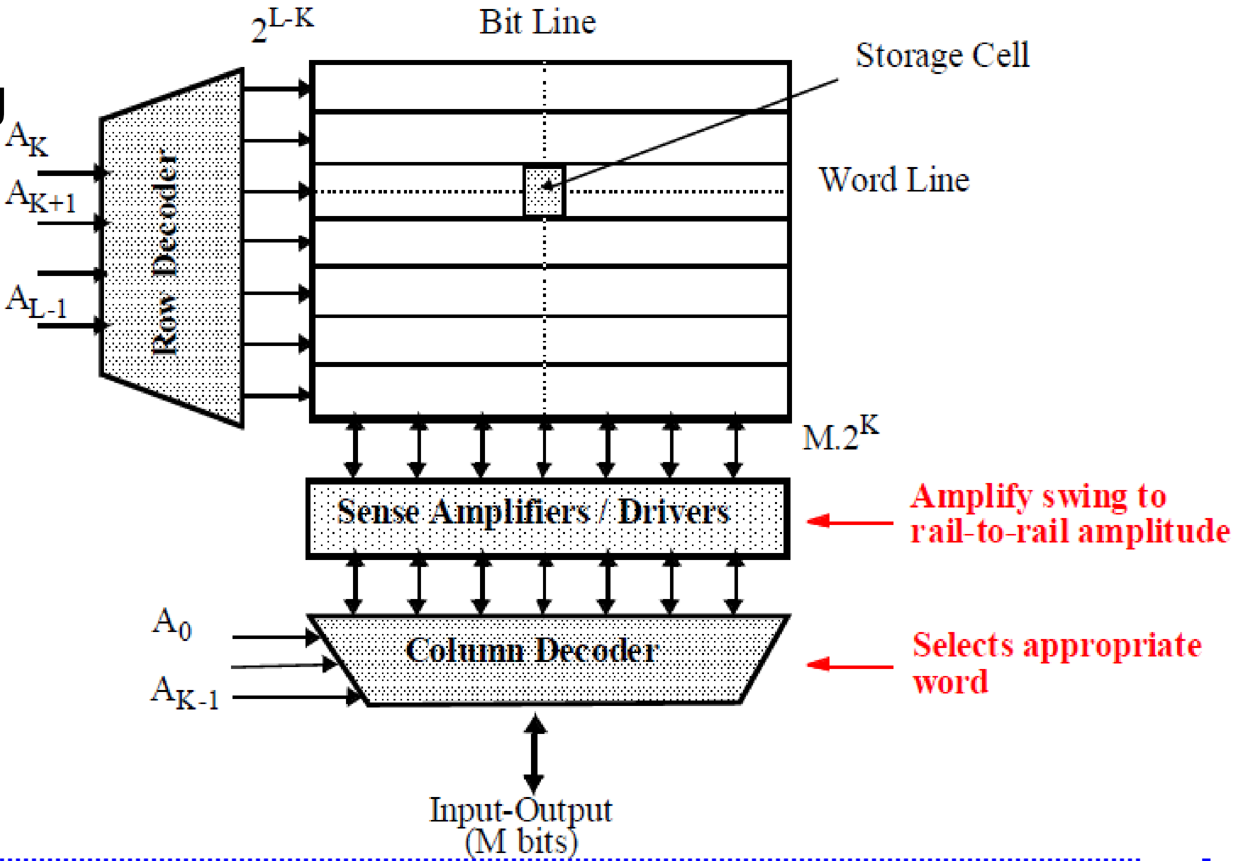


Memory Components

Row decoder
Cell array
Sense amplifier
Column multiplexing

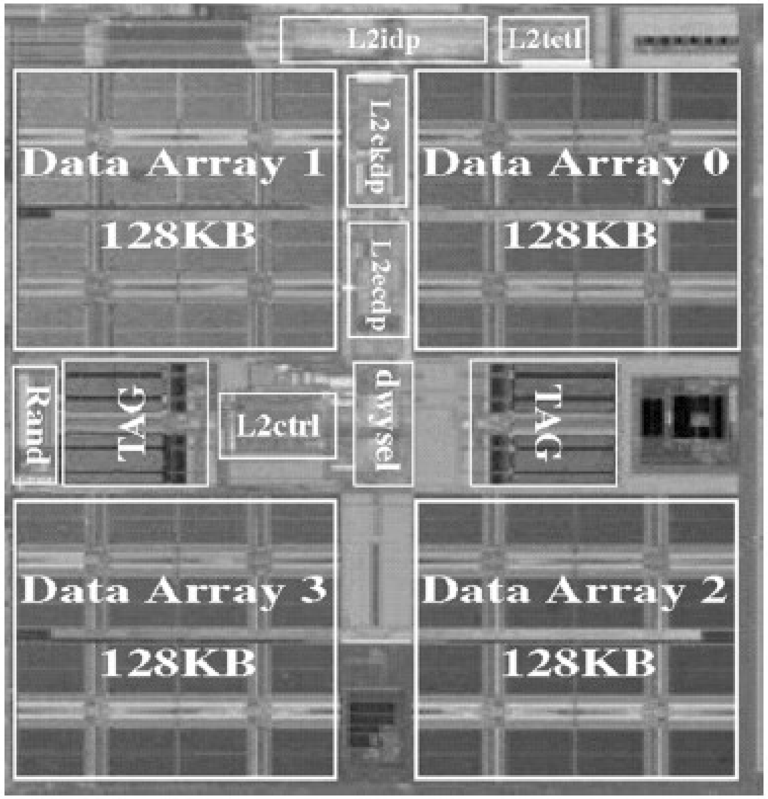


Problem: ASPECT RATIO or HEIGHT >> WIDTH



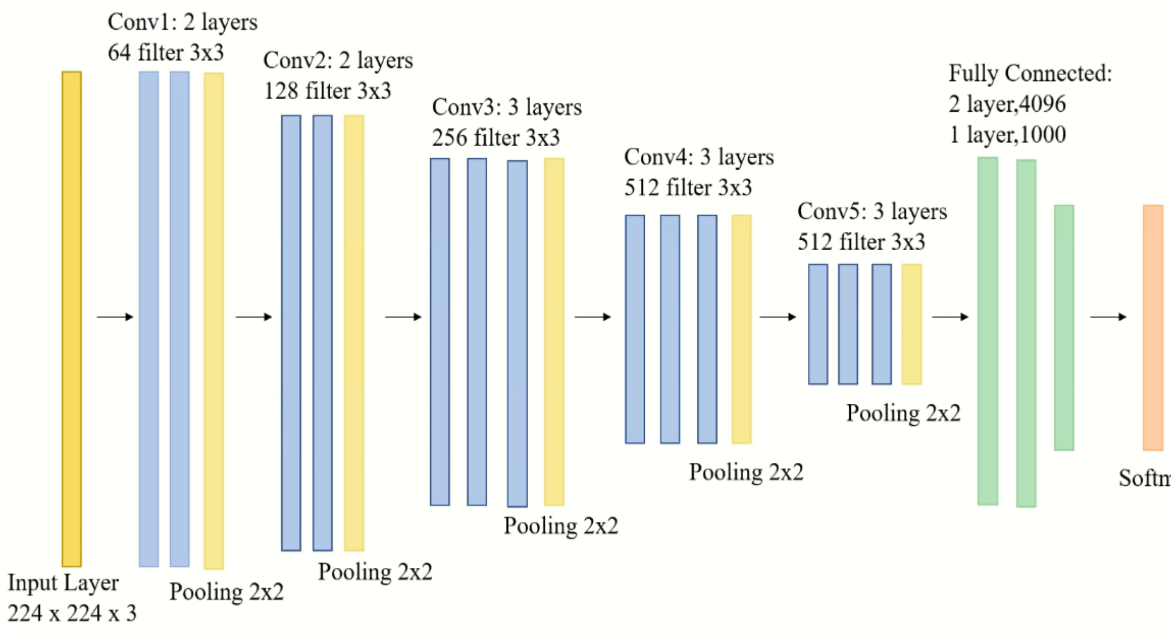
Large SRAMs

- Large SRAMs are split into subarrays for speed
  - typically 128 or 256 words (rows) per vertical bit-line (column)
  - typically 128 or 256 bits (columns) on each horizontal word-line (row)
- Ex: UltraSparc 512KB cache
  - 4 * 128 KB subarrays
  - each subarray has 16 * 8KB banks
  - 256 rows x 256 cols / 8KB-bank



Down-Sampled Input Image

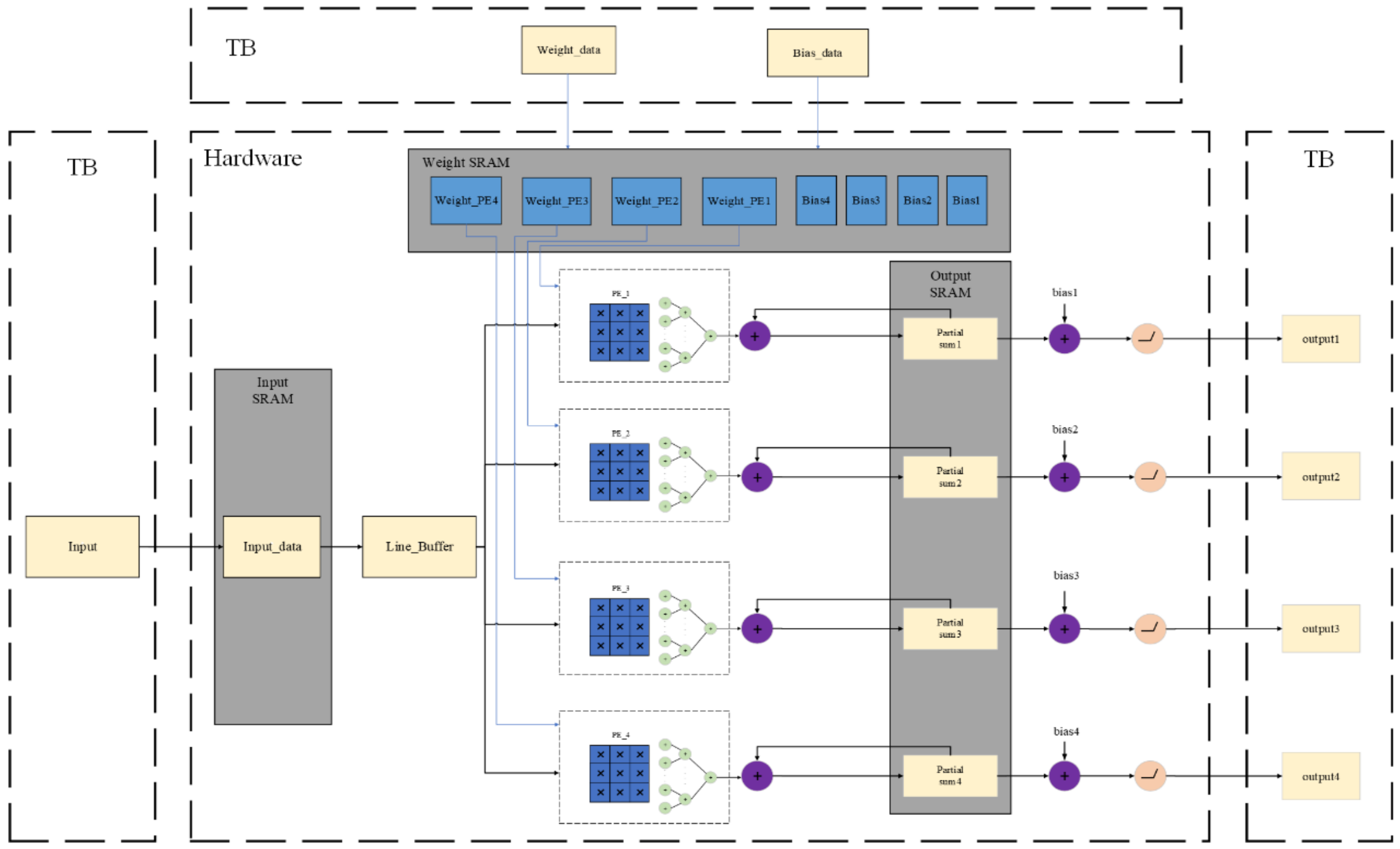
- original input image size
  - 224x224
- down-sampled input image size to 56x56 in testbench
  - 1/16 image size
  - 1/16 computation
  - smaller execution time
- consider only 1st VGG-16 layer
  - 3 input channels
  - 64 output channels



# of input channels	# of output channels	Input featuremap size	Filter Kernel size
3	64	56x56	3x3

Synthesis of SRAM

- in HW5, Input/Weight/Output SRAM units are in testbench
- in HW6, they are real hardware generated from memory ccompiler
- only execute the 1st VGG-16 layer with ICP=1, KWP=9, and OCP=4

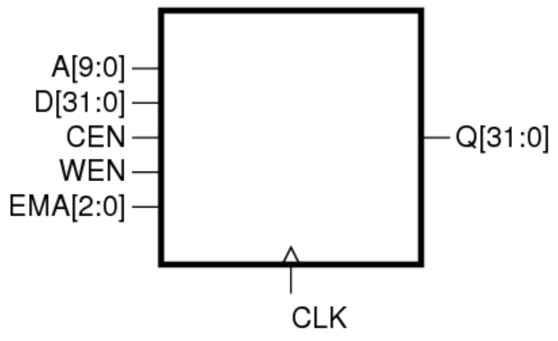


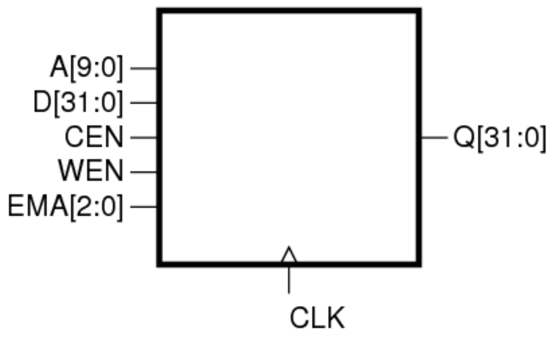
TSMC 90nm Memory Compiler

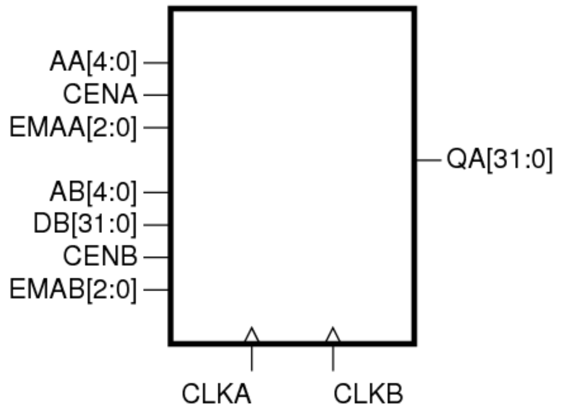
- 90nm

TSMC_90nm	Register file	SRAM
Single-port	RF_SP_ADV	SRAM_SP_ADV
Two-port	RF_2P_ADV	-
Dual-port	--	SRAM_DP_ADV

single-port, two-port , dual-port



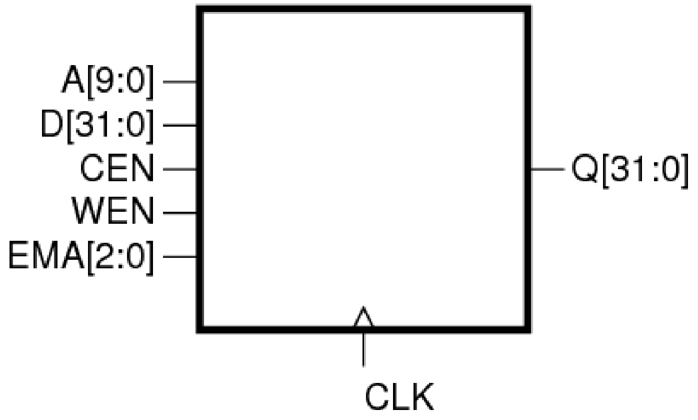




TSMC_40nm	Register file	SRAM
Single-port	RF_SP_HDE (rvt_hvt_rvt) RF_SP_HSD (rvt_rvt_hvt)	SRAM_SP_HDE (rvt_hvt_rvt) SRAM_SP_HSC (rvt_hvt_rvt)
Two-port	RF_2P_HSE (rvt_hvt_rvt)	-
Dual-port	--	SRAM_DP_HDE (rvt_hvt_rvt)

TSMC 90nm Single-Port Pins

- chip-enable (CEN) and write-enable (WEN) are active-low



Pin	Description
A[9:0]	Addresses (A[0] = LSB)
D[31:0]	Data Inputs (D[0] = LSB)
CLK	Clock
CEN	Chip Enable (active low)
WEN	Write Enable (active low)
Q[31:0]	Data Outputs (Q[0] = LSB)
EMA[2:0]	Extra Margin Adjustment (EMA[0] = LSB)

TSMC 40nm Single-Port Pins

- contain pins related to design-for-testability (DFT)
- set initial values of these pins for normal operations

Pin	Descrption
CEN	Chipe Enable (active low)
WEN	Write Enable (active low)
A	Addresses(A[0]=LSB)
D	Data Inputs (D[0]=LSB)
Q	Data Outputs (Q[0]=LSB)
CLK	Clock
WENY	Multiplexor out (WEN CEN A D)
CENY	
AY DY	

Pin	Descrption
EMA	Extra Margin Adjustment
EMAW	
EMAS	
BEN	Bypass mode,active low
TEN (enable)	TEST MODE,active low (CEN WEN A D Q 同前面意思)
TCEN	
TWEN	
TA TD TQ	
RET1N	Retention mode, acitve low
STOV	Synchronous clock enable, acitve low

TSMC 90nm Memory: Register-File

- register-file-based memory

single-port

two-port

Single-Port 90nm Register File rf_sp_adv		
Parameter	Ranges	
Numbers of words	Mux=1	8 to 128
	Mux=2	16 to 256
	Mux=4	32 to 512
Numbers of bits	Mux=1	8 to 128
	Mux=2	4 to 128
	Mux=4	2 to 64
Total memory bits	Mux=1	64 to 16,384 bits
	Mux=2, 4	64 to 32,768 bits

Two-Port 90nm Register File rf_2p_adv		
Parameter	Ranges	
Numbers of words	Mux=1	8 to 128
	Mux=2	16 to 256
	Mux=4	32 to 512
Numbers of bits	Mux=1	2 to 128
	Mux=2	2 to 64
	Mux=4	2 to 32
Total memory bits	16 to 16,384 bits	

# of rows = # of words
# of columns = # of bits * mux

TSMC 90nm Memory: SRAM

- sram-based memory

single-port

dual-port

Single-Port 90nm SRAM sram_sp_adv		
Parameter	Ranges	
Numbers of words	Mux=8	256 to 4096
	Mux=16	512 to 8192
	Mux=32	1024 to 16384
Numbers of bits	Mux=8	2 to 128
	Mux=16	2 to 64
	Mux=32	2 to 32
Total memory bits	512 to 524,288 bits	

Dual-Port 90nm SRAM sram_dp_adv		
Parameter	Ranges	
Numbers of words	Mux=4	128 to 2048
	Mux=8	256 to 4096
	Mux=16	512 to 8192
Numbers of bits	Mux=4	2 to 128
	Mux=8	2 to 64
	Mux=16	2 to 32
Total memory bits	256 to 262,144 bits	

Memory Synthesis in Synopsys DC

- use memory compiler to generate .lib and .v files
  - select register-file or sram
  - select single-port, two-port, or dual-port
  - specify # of words, # of bits, mux_width
  - give module name (e.g., sram_sp_1024x32)
- add the memory modules into your design
  - first convert .lib into .db file for Synopsys DC synthesis
  - use module instantiation to add memory modules
  - combine with your other RTL codes, e.g., in top.v
- use Synopsys DC to synthesize the entire ckt.
  - synthesized ckt., e.g., in top_gate.v
- simulate with all .v files