EEDG/CE 6303: Testing and Testable Design

Mehrdad Nourani

Dept. of ECE
Univ. of Texas at Dallas

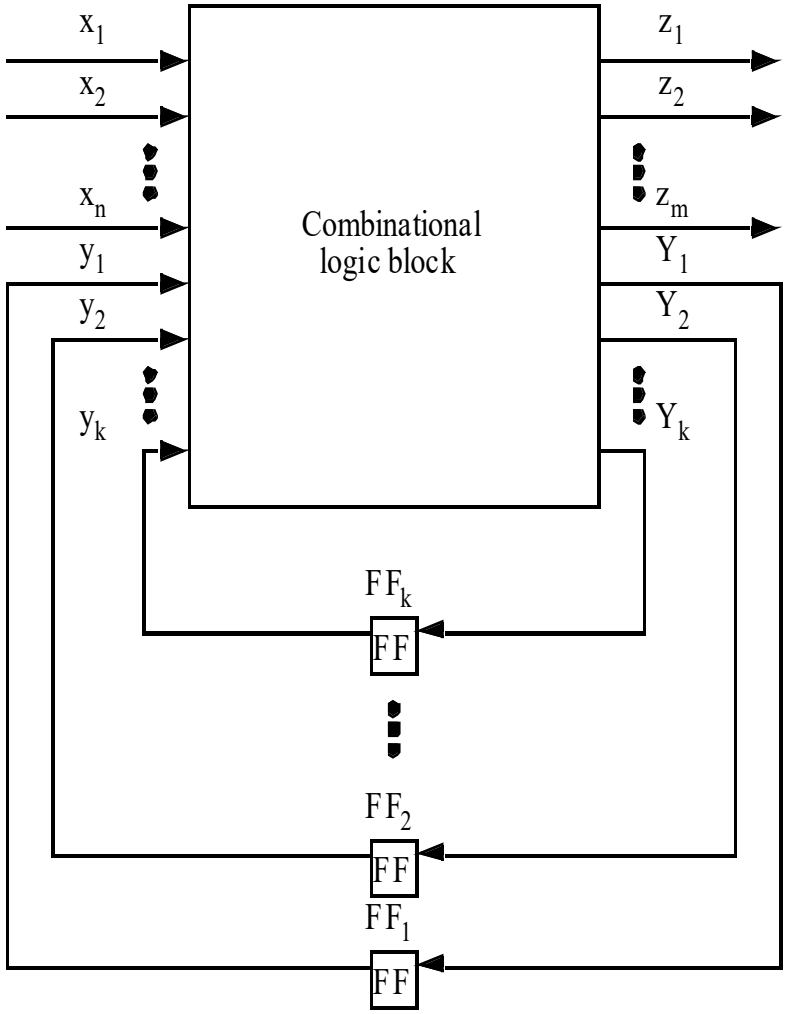
Session 09

Scan Design for Testability

Basic Concept

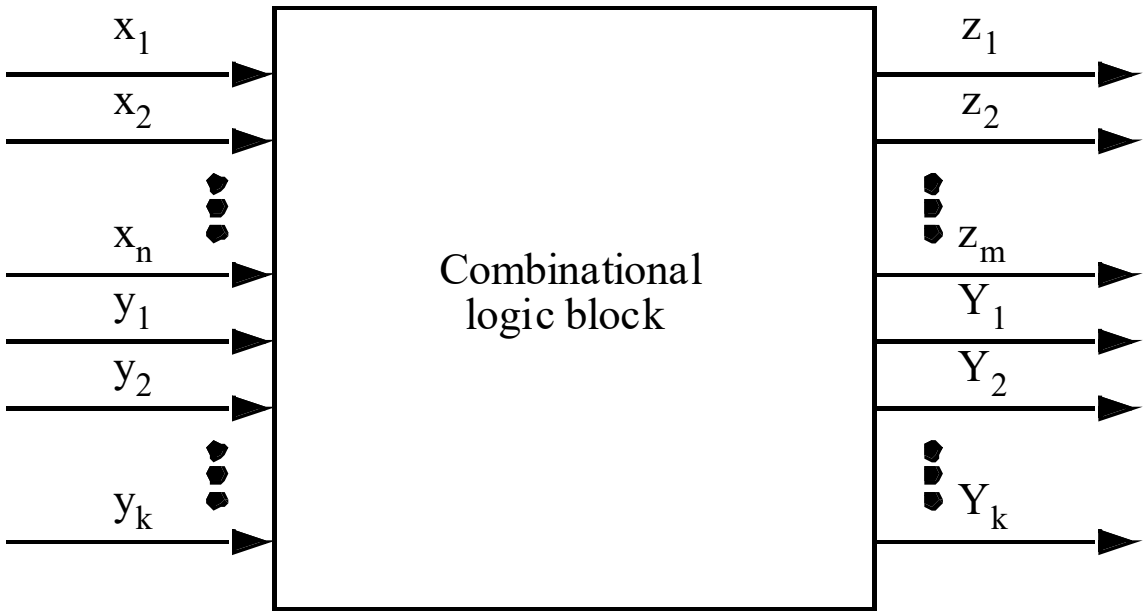
Architectural View

- A sequential circuit is viewed as
  - —A combinational logic block, with
    - Primary inputs $x_1, x_2, \ldots, x_n$
    - Primary outputs $z_1, z_2, \ldots, z_m$
    - State inputs (present state) $y_1, y_2, \ldots, y_k$
    - State outputs (next state) $Y_1, Y_2, \ldots, Y_k$
  - —Y_i and y_i are respectively the input and output of FF_i



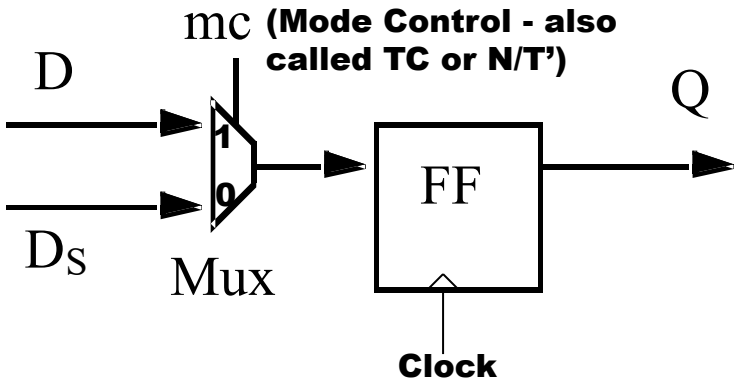
Test Generation in Scan

- Hence, for purposes of test development
  - State inputs treated as primary inputs
  - State outputs treated as primary outputs
- By reducing test generation for sequential circuits to test generation for combinational circuits, scan
  - Reduces test development cost
  - In many cases, enables attainment of acceptable fault coverage



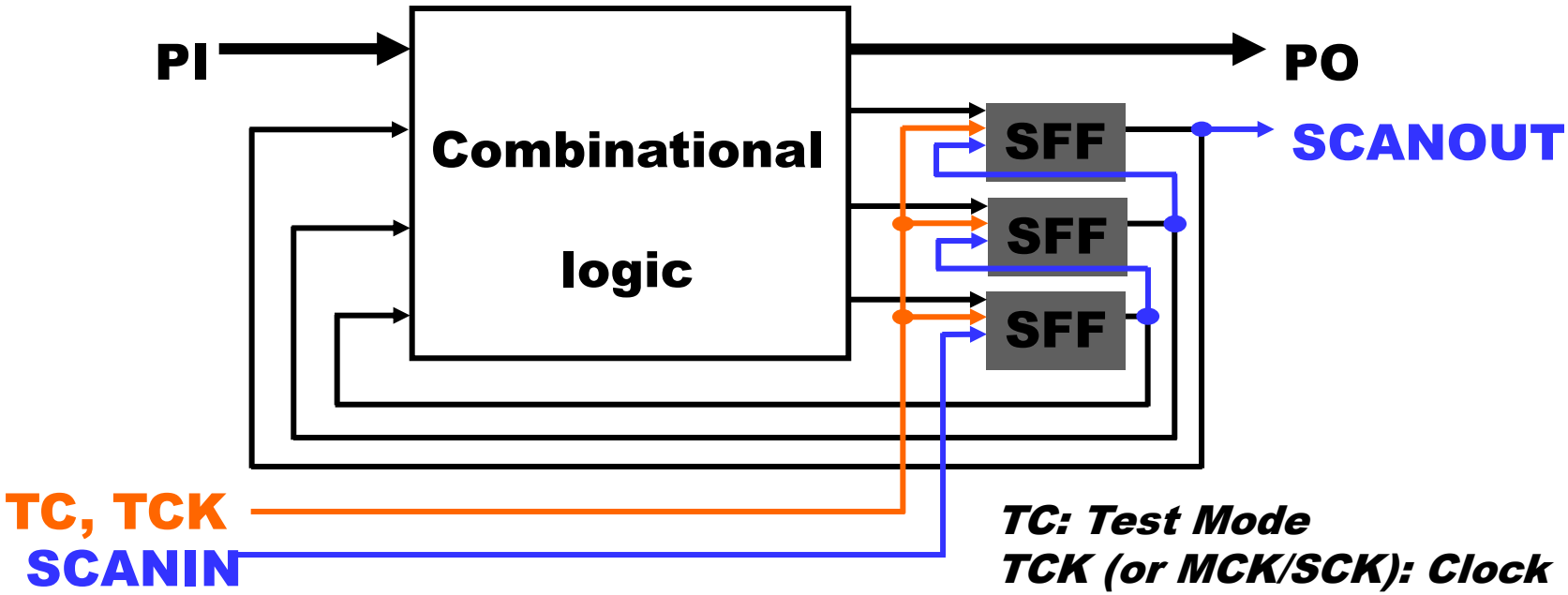
Scan as a DFT Method

- In scan DFT (design for testability) methodology, flip-flops (or latches) designed to support two modes
  - **Normal mode**: Flip-flops configured as in the original circuit
  - **Test mode**: Flip-flops configured as one or more shift-registers, called **scan registers** or **scan chains**
- Most Common Scan Flip-Flop (SFF) cell



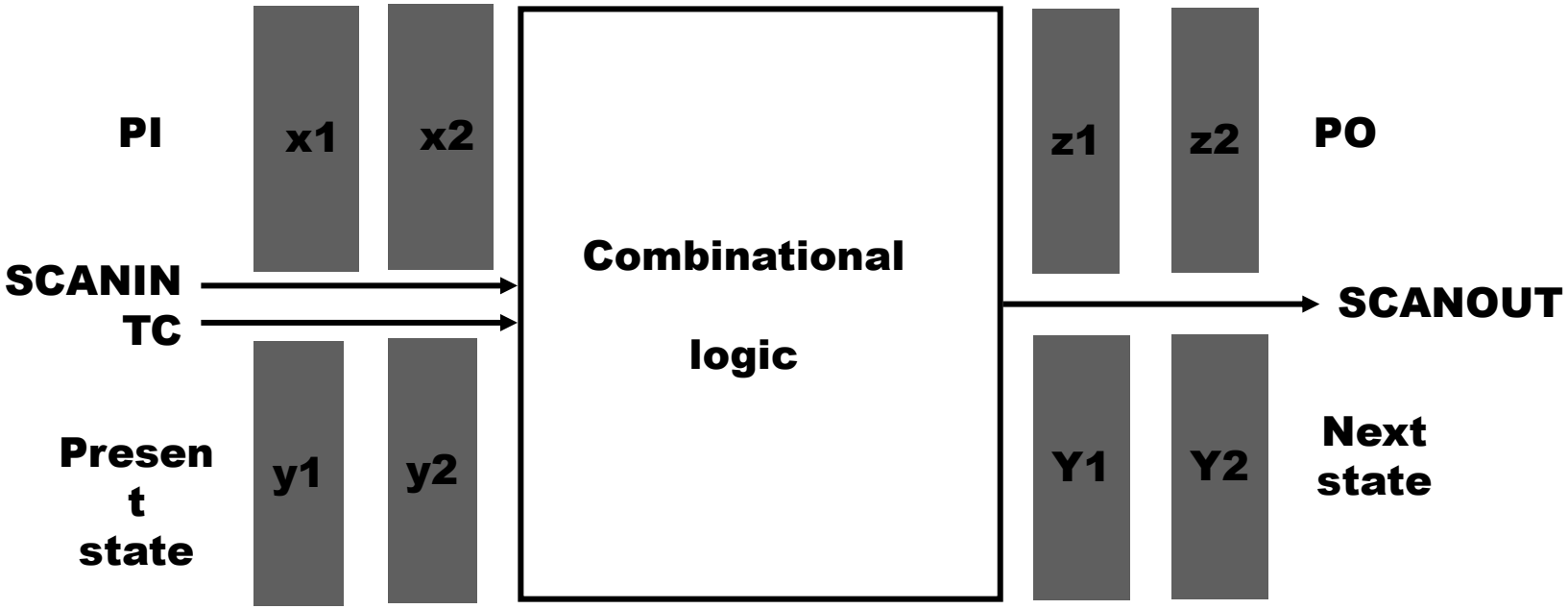
Scan Structure

- Testing using scan
  - In normal mode, responses at state outputs captured in flip-flops
  - Circuit then configured in the test mode
    - Scan registers clocked
    - The output of the last flip-flop in scan chain observed
  - At the same time, values to be applied at state inputs in the subsequent test shifted into flip-flops



Applying Test in Scan

- **Shift Register Test:** TC=0 (scan mode) and apply the 001100…
  - Test length: $n_{sff}+4$
- **Scan Test:**
  - TC=0 (scan mode)
  - Shift yi values in FFs.
  - Put xi values in PIs.
  - TC=1 (normal mode) and check zi's.
  - Apply clock to TCK (still normal mode).
  - TC=0 (scan mode) and shift Yi's out for check.



Applying Test in Scan (cont.)

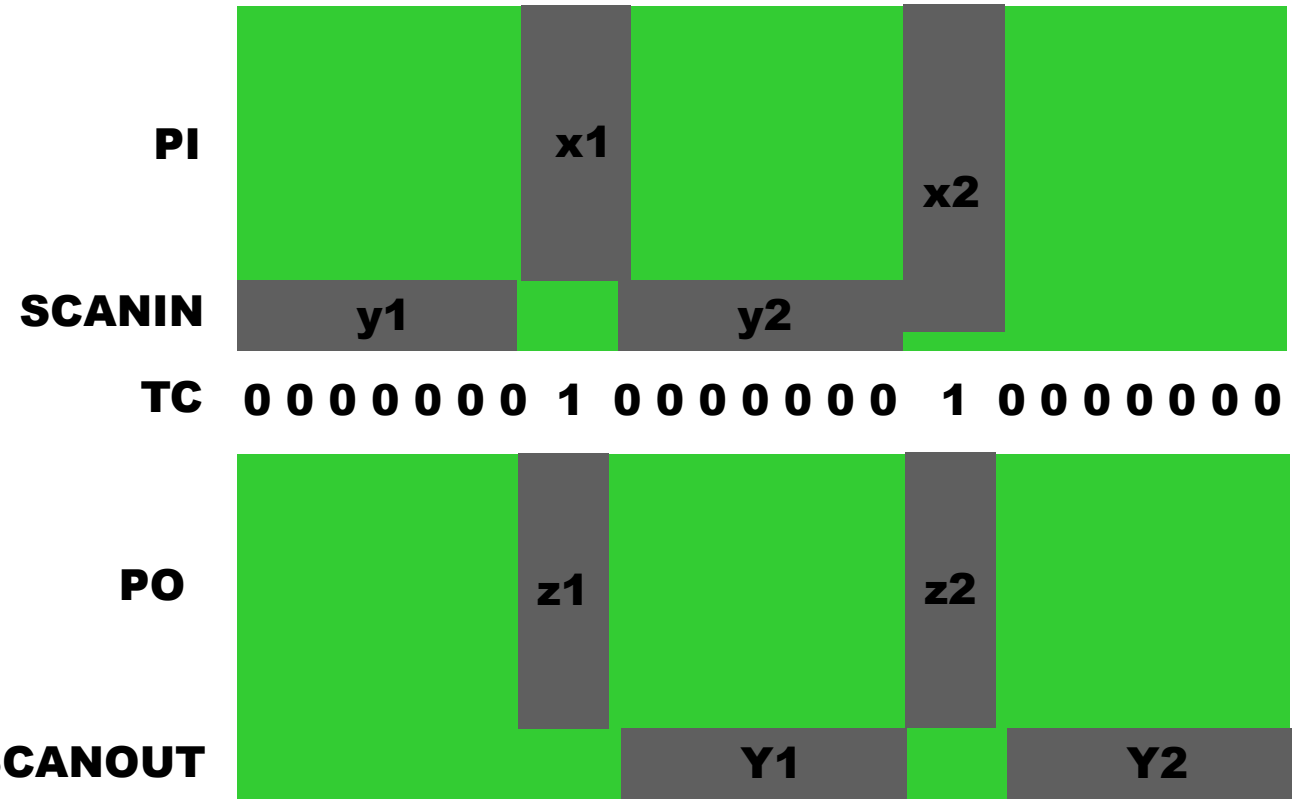
- **Total scan test length (number of clocks)**

  $(n_{sff}+4)+(n_{sff}+1)\ n_{comb} + (n_{sff}-1) = (n_{comb}+2)n_{sff}+n_{comb}+3$

  — n_{comb} = number of combinational vectors

  — n_{sff} = number of scan flip-flops

- Without considering shift register test: $(n_{sff}+1)\ n_{comb} + (n_{sff}-1)$



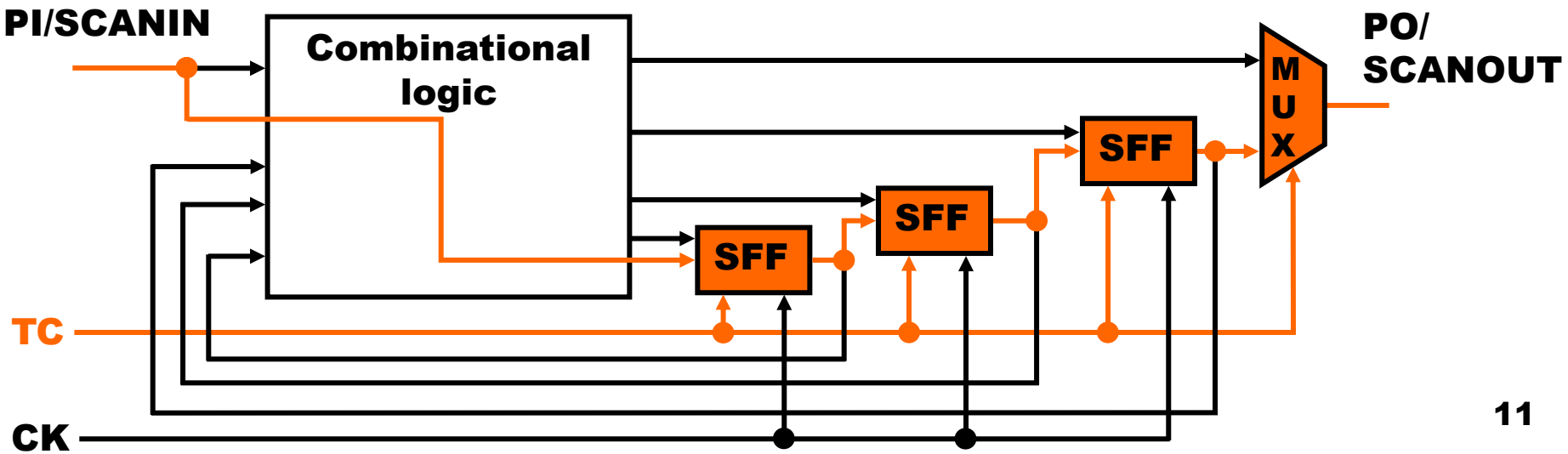
Applying Test in Scan (cont.)

- Scan register must be tested prior to application of scan test sequences.
- A shift sequence 00110011 . . . of length n_{sff}+4 in scan mode (TC=0) produces 00, 01, 11 and 10 transitions in all flip-flops and observes the result at SCANOUT output.
- Total scan test length:

  $(n_{comb} + 2)\, n_{sff} + n_{comb} + 3$ *clock periods*.
- Example: 2,000 scan flip-flops, 500 comb. vectors, total scan test length ~ 10^6 clocks.
- Multiple scan registers reduce test length.

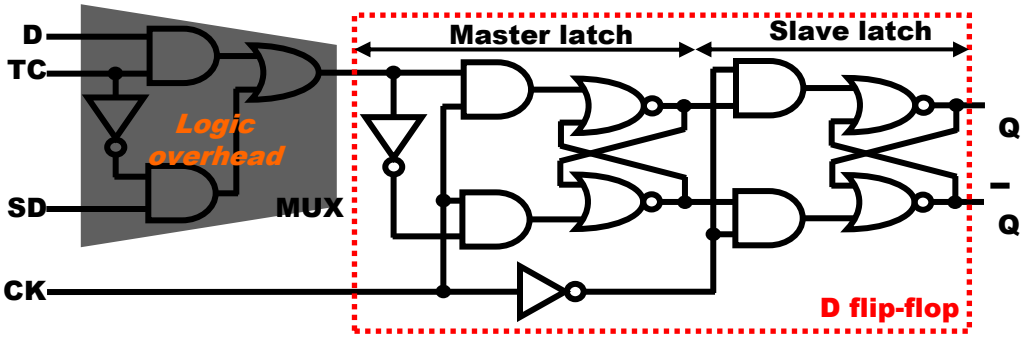
Multiple Scan Registers

- Scan flip-flops can be distributed among any number of shift registers, each having a separate *scanin* and *scanout* pin.
- Test sequence length is determined by the longest scan shift register.
- Just one *test control* (TC) pin is essential.



Scan Test Overhead

- IO pins: One pin necessary.
- Area overhead:
  - *Gate overhead* = [4 $n_{sff}/(n_g+10n_{ff})$] x 100%
    - n_g = *comb. Gates*
    - n_{ff} = *flip-flops*
    - Example – n_g = 100k gates, n_{ff} = 2k *flip-flops*, overhead = 6.7%.
  - More accurate estimate must consider scan wiring and layout area.
- Performance overhead:
  - Multiplexer delay added in combinational path; approx. two gate-delays.
  - Flip-flop output loading due to one additional fanout; approx. 5-6%.



Timing and Power During Scan

- Small delays in scan path and clock skew can cause race condition.
- Large delays in scan path require slower scan clock.
- Dynamic multiplexers: Skew between TC and TC signals can cause momentary shorting of D and SD inputs.
- Random signal activity in combinational circuit during scan can cause excessive power dissipation.

Full-Scan vs. Partial Scan

- In a full-scan circuit, scan mode and scan chains enable
  - Desired values to be **scanned in** into every flip-flop from *ScanIn*
  - Response captured in every flip-flop to be **scanned out** and observed at *ScanOut*
- In a partial-scan, a subset of flip-flops is scanned.
  - Minimize area overhead and scan sequence length, yet achieve required fault coverage
  - Exclude selected flip-flops from scan:
    - Improve performance
    - Allow limited scan design rule violations
  - Shorter scan sequences
- Hence, combinational part of the circuit used to generate tests

Effect on Cost and Performance

- Cost and benefits of scan
  - —Area overhead: Increase in circuit area to add extra circuitry, route additional control signals and/or clocks
    - – Increases chip area and hence cost, since
      - + Fewer chips manufactured per wafer
      - + Increase in area decreases yield
  - —Performance penalty: Additional logic typically increases delay in normal mode
    - – Performance penalty may be reduced at the cost of higher area

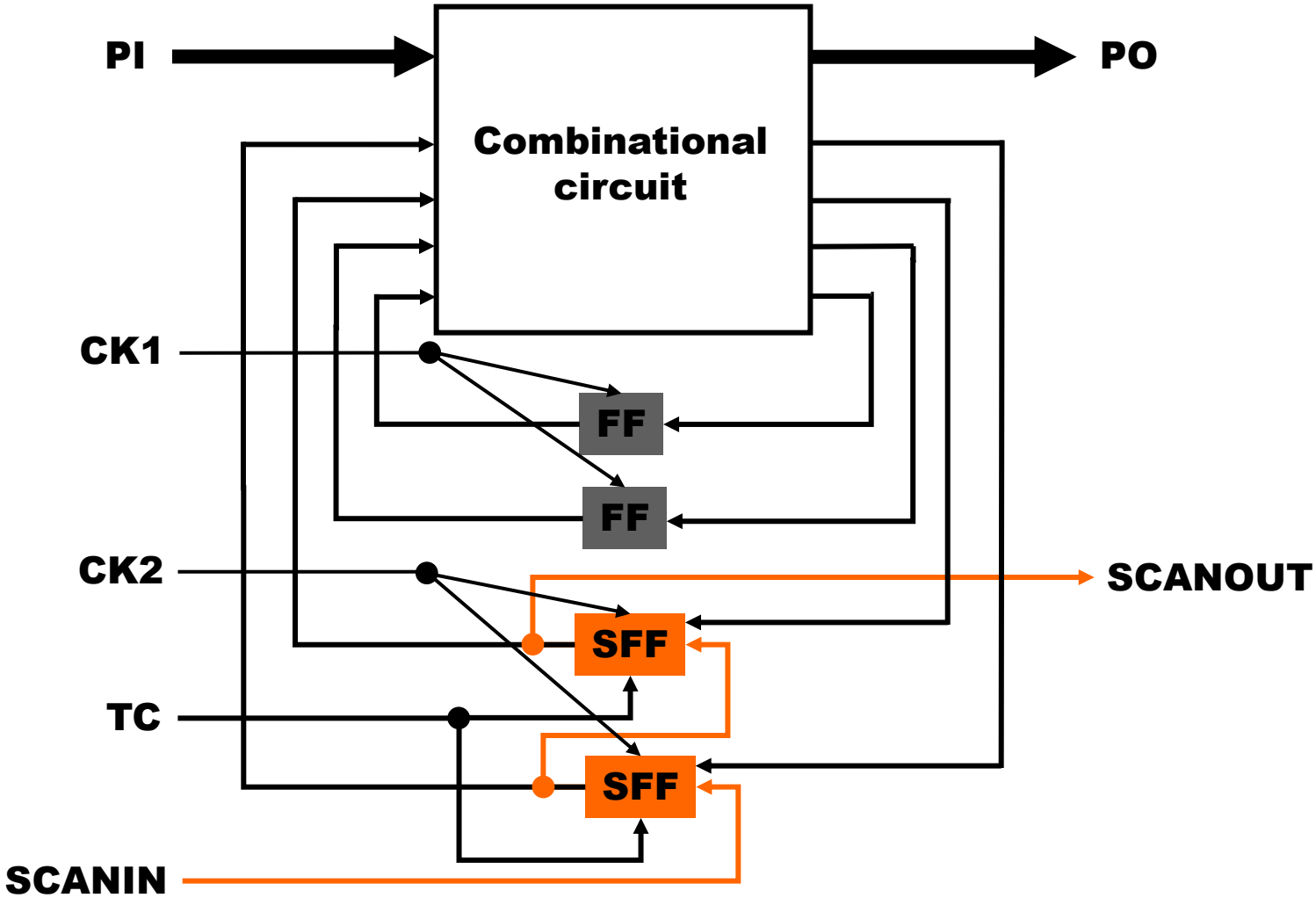
Effect on Cost and Performance (cont.)

- Cost and benefits of scan
  - Need extra pins for ScanIn, ScanOut, control inputs, and so on
  - Typically increase in test application time
  - For many sequential circuits, the only way to achieve acceptable fault coverage
  - Useful for debugging first silicon, i.e., the first batch of chips fabricated for a given design
  - Used to locate failing components when an operational system fails

Partial Scan

- Overheads of scan can be reduced by replacing only a subset of flip-flops in a circuit by scan flip-flops, i.e., via partial scan
- How to select flip-flops to scan?
  - To reduce area overhead, select a minimum number of flip-flops
  - To reduce performance penalty, avoid selecting flip-flops along critical paths, i.e., paths whose delays are equal to or close to the clock period
  - However, flip-flops should be selected to ensure improvements in controllability and observability values so as to provide acceptably high fault coverage

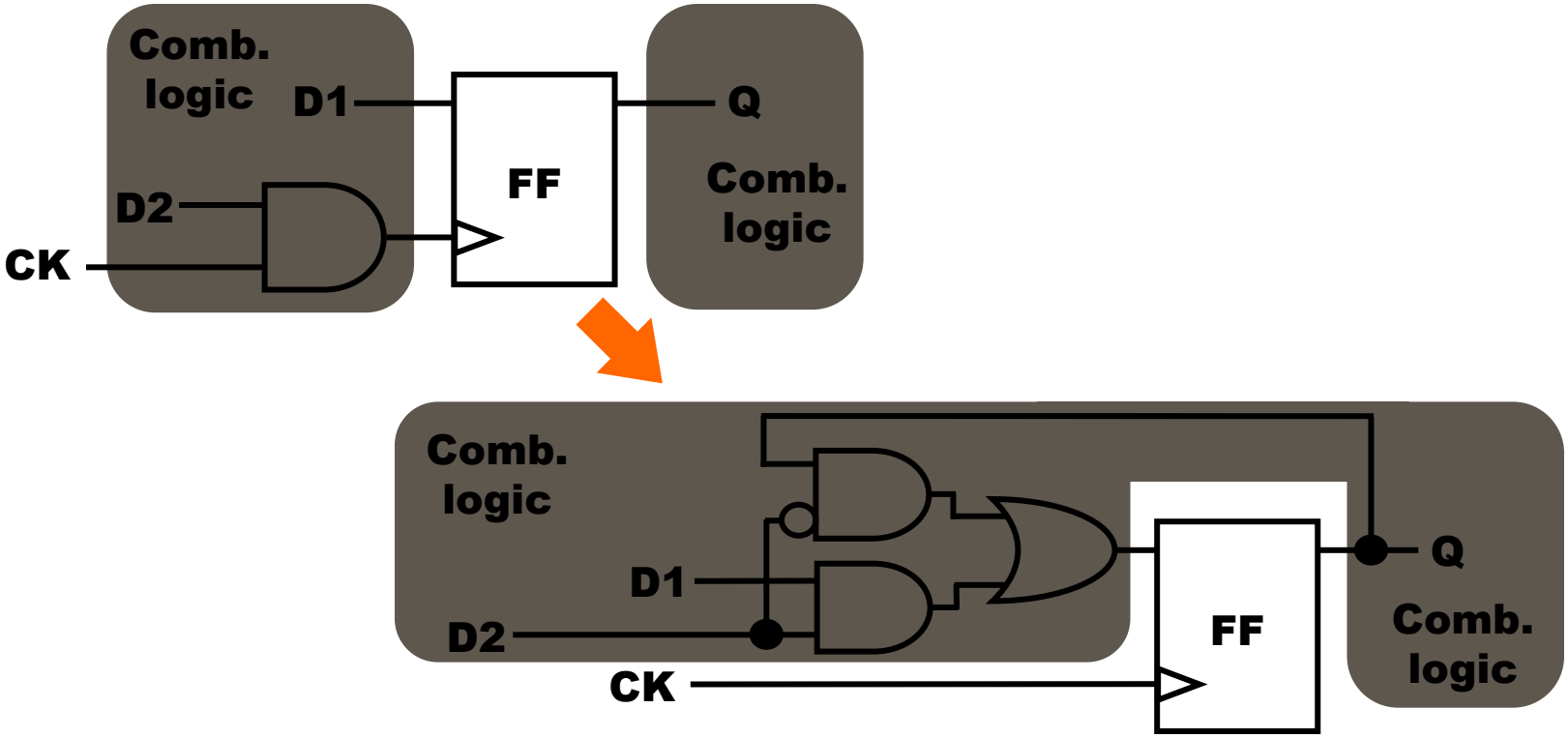
Partial Scan Architecture



Scan Cells

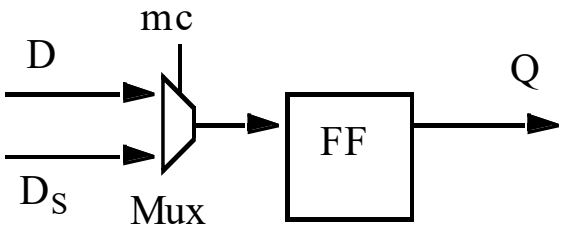
Scan Design Rules

- Use only clocked D-type of flip-flops for all state variables.
- At least one PI pin must be available for test; more pins, if available, can be used.
- All clocks must be controlled from PIs.
- Clocks must not feed data inputs of flip-flops.



Basic Scan Cell

- **Multiplexed-input scan**
  - —Flip-flop design must be modified such that
    - In normal mode, the input to FF_i is the value at next state output Y_i
    - In the test mode, the input to FF_i is the value at the output of the previous flip-flop in the scan chain
  - —This can be achieved by adding a multiplexer
    - Two data inputs, D and D_S
    - Mode control input *mc*
      - When *mc = normal (e.g. 0)*, input D is selected
      - When *mc = test (e.g. 1)*, input D_S is selected

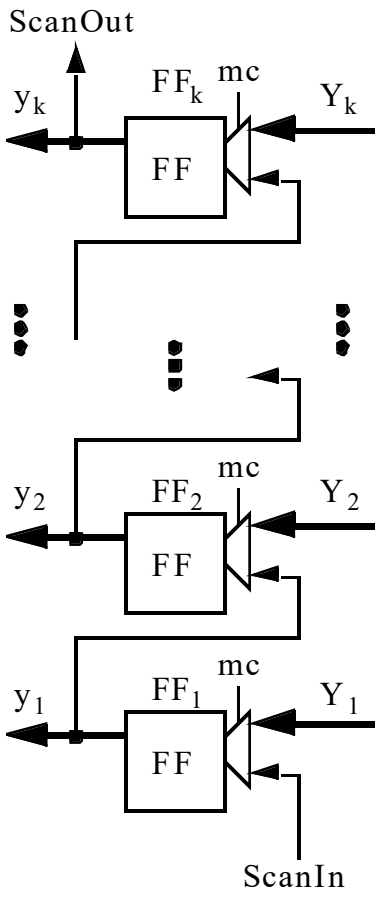


Basic Scan Cell (cont.)

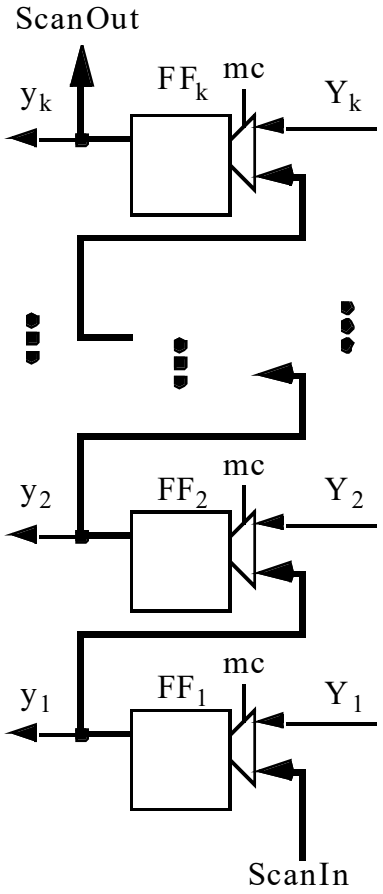
- Scan flip-flops are connected to support test and scan modes
  - The D input of FF_i is connected to the state output Y_i
  - The D_S input of FF_i is connected to the output of the previous flip-flop in the scan chain
  - The D_S input of the first flip-flop in the scan chain is connected to *ScanIn*, the input to the scan chain
  - The output of FF_i is connected to the state input y_i
  - The output of the last flip-flop in the chain is connected to *ScanOut*, the output of the scan chain

Normal vs. Scan Mode

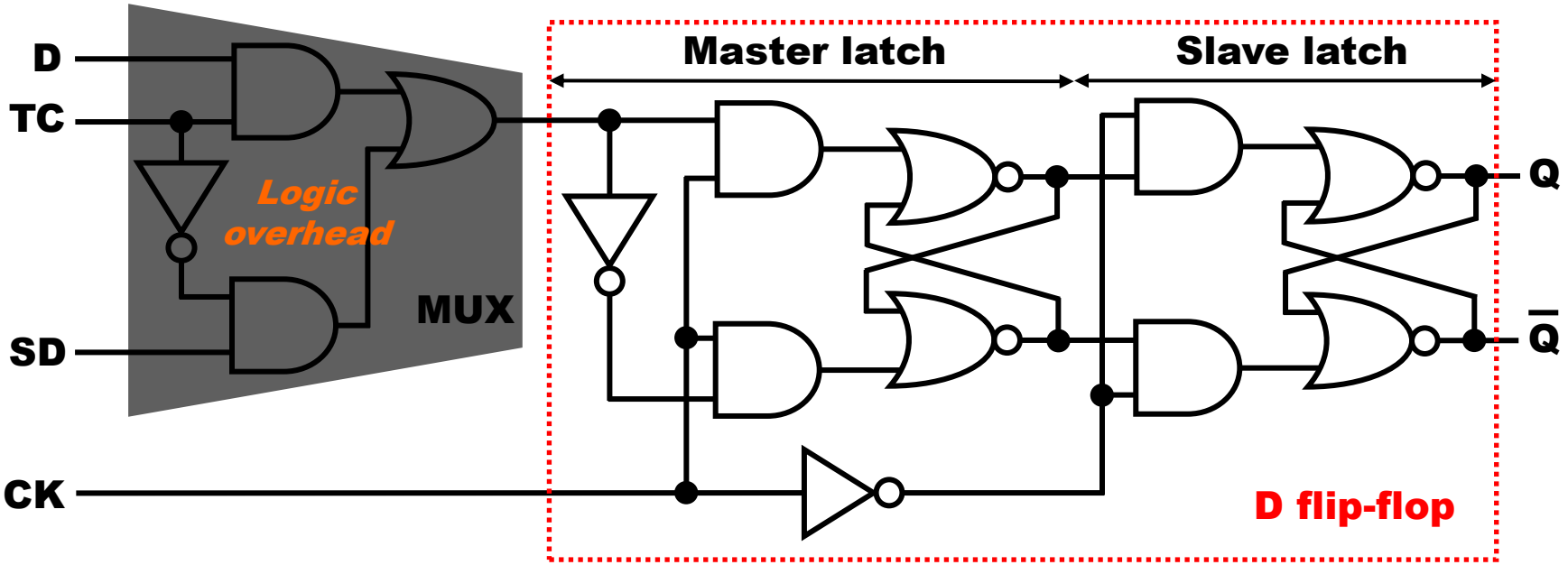
Normal Mode

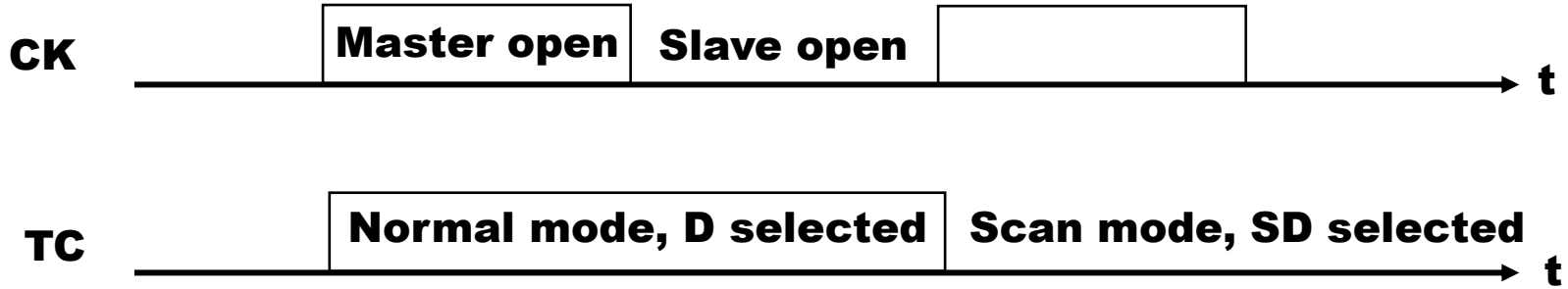


Scan Mode

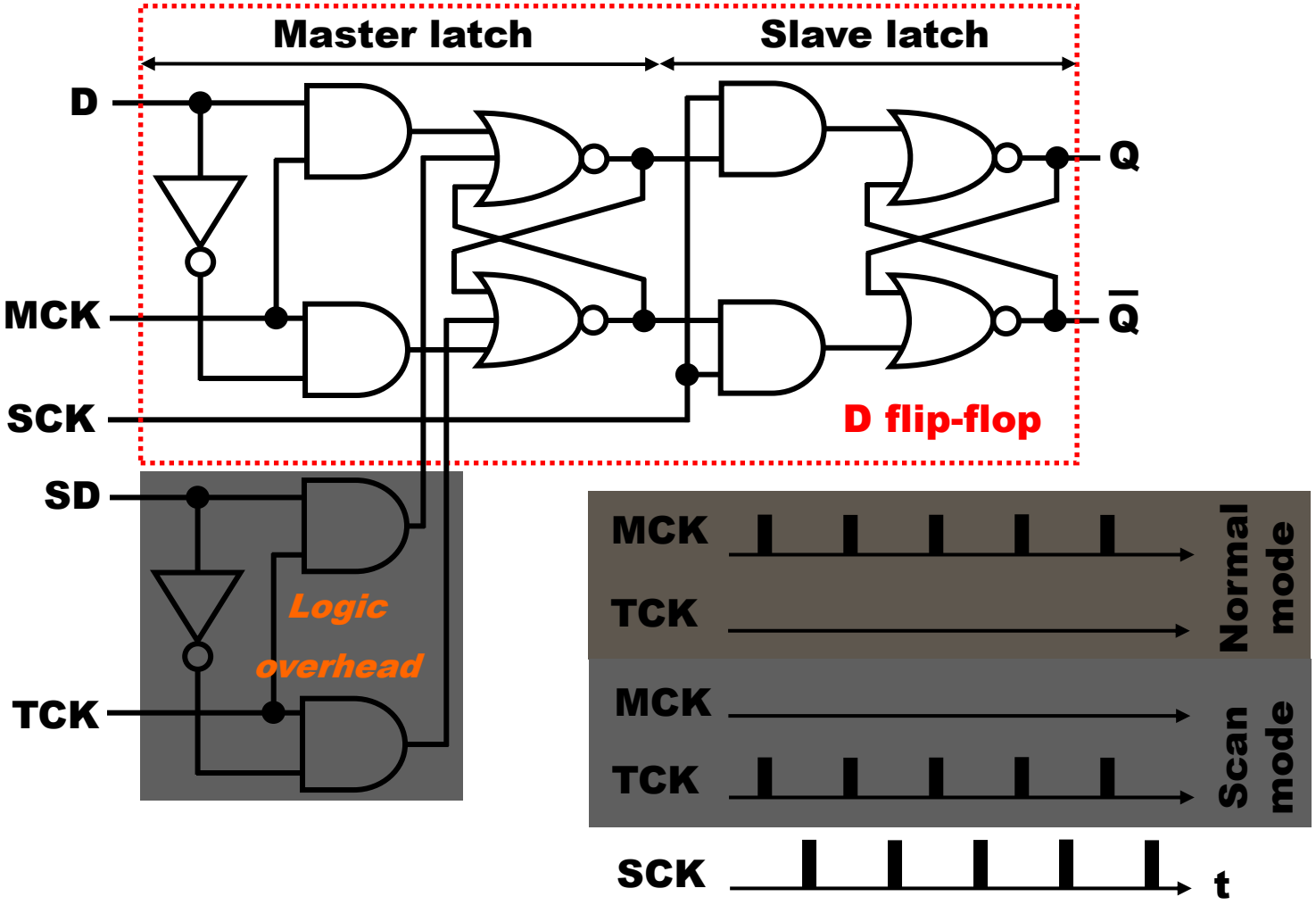


Implementation of Scan FF Cell



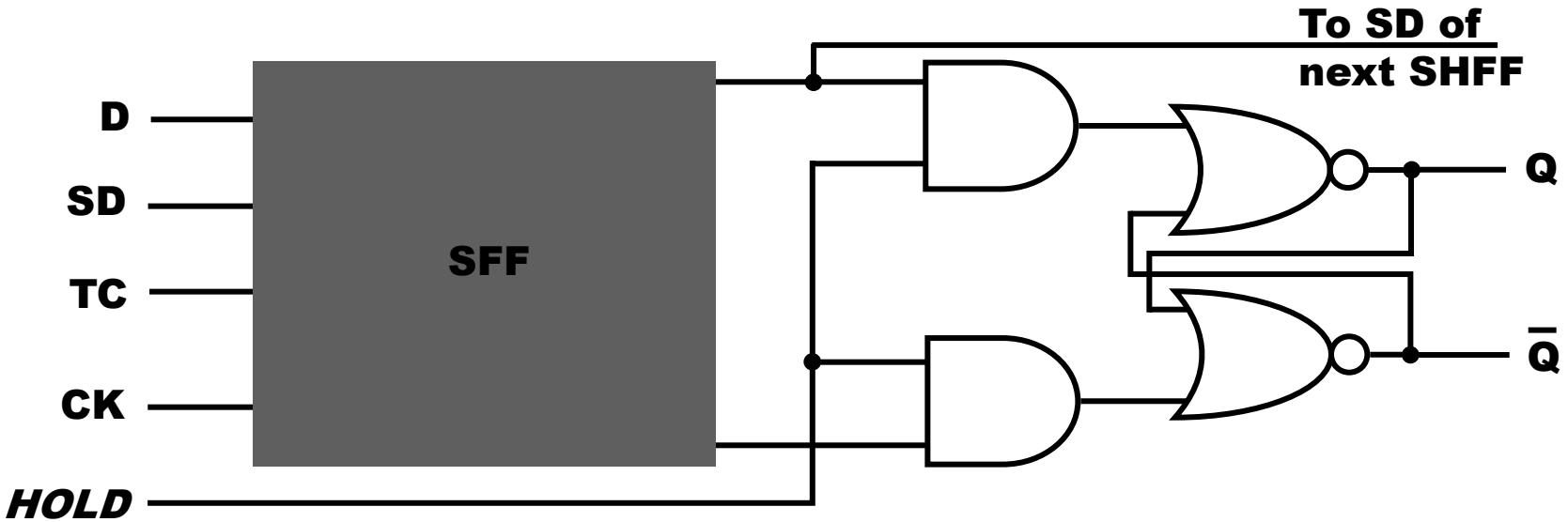


Level-Sensitive Scan D Latch (LSSD)

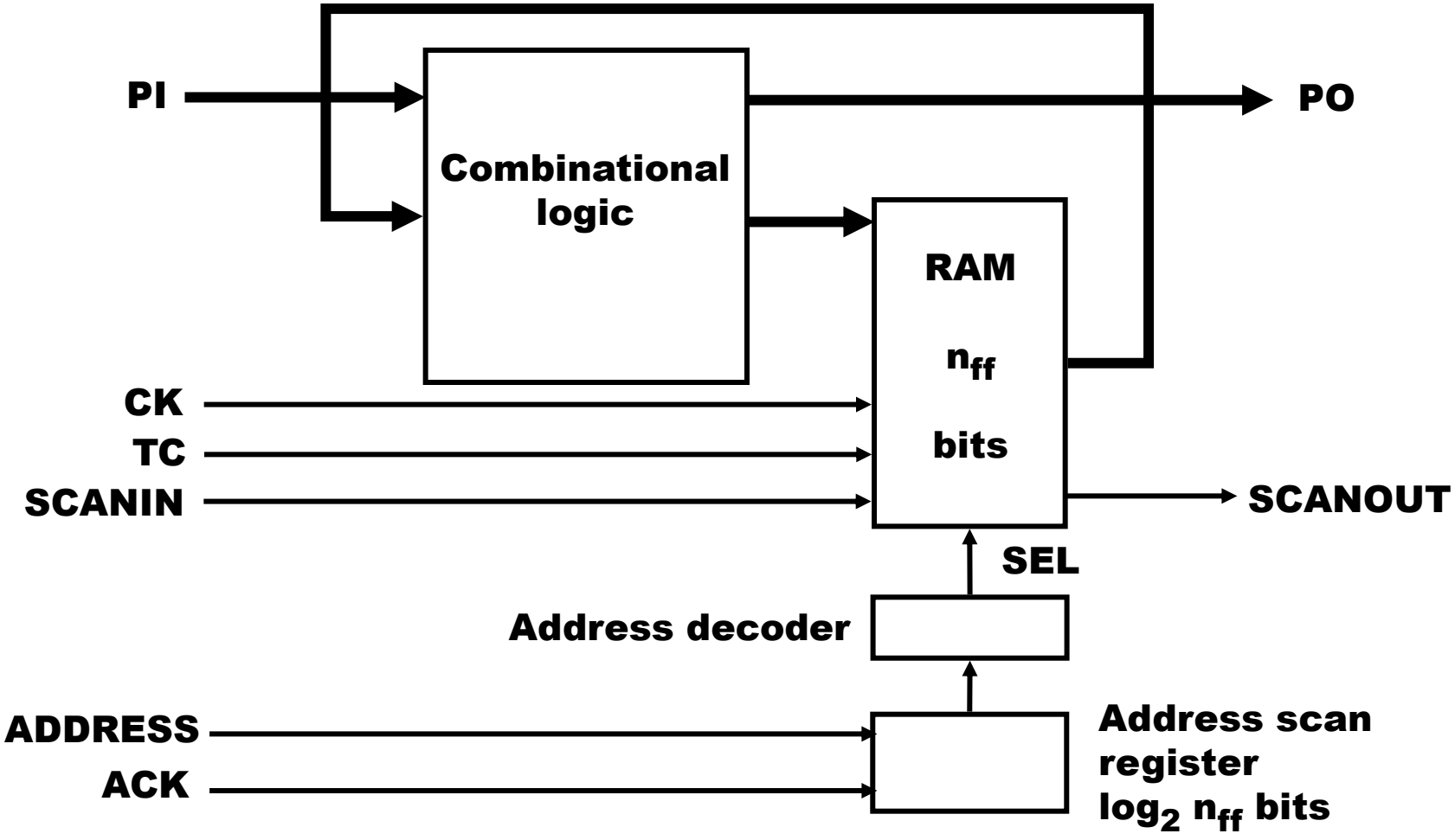


Scan-Hold Flip Flop (SHFF)

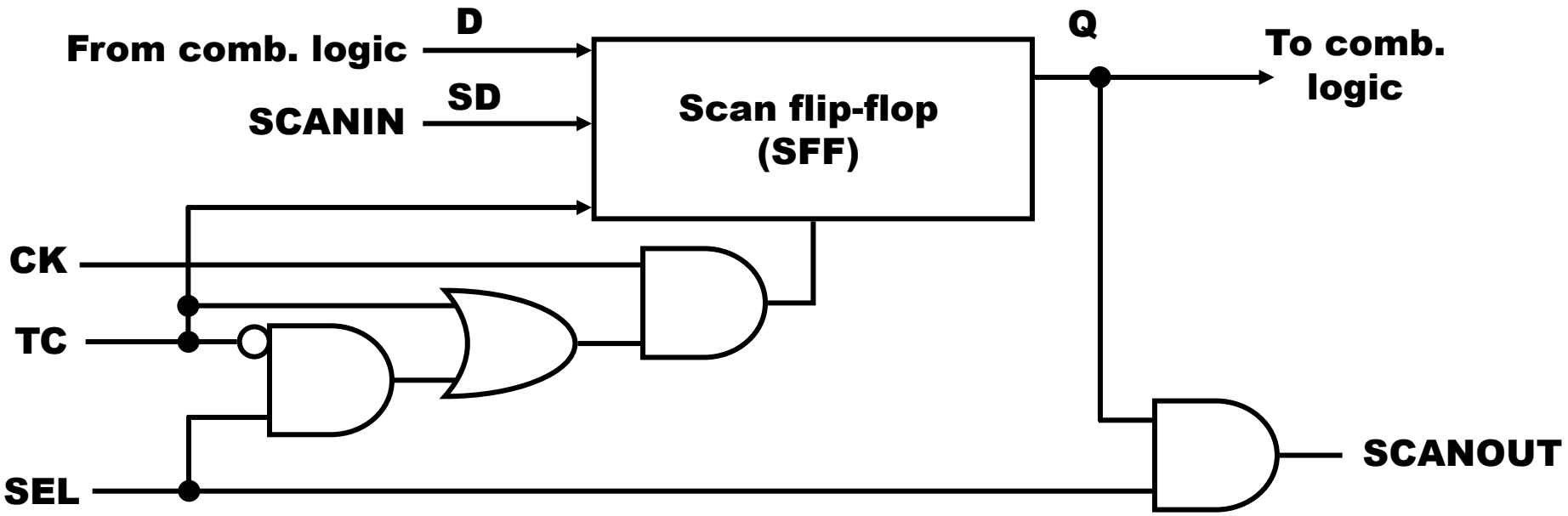
- The control input *HOLD* keeps the output steady at previous state of flip-flop.
- Applications:
  - Reduce power dissipation during scan
  - Isolate asynchronous parts during scan test
  - Delay testing (applying V1→V2 pair)



Random-Access Scan (RAS)

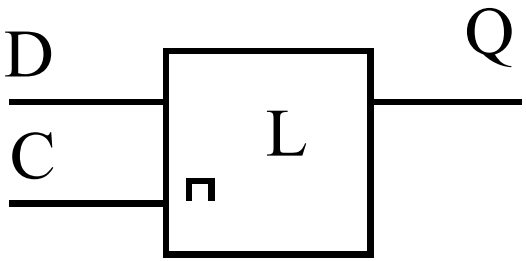


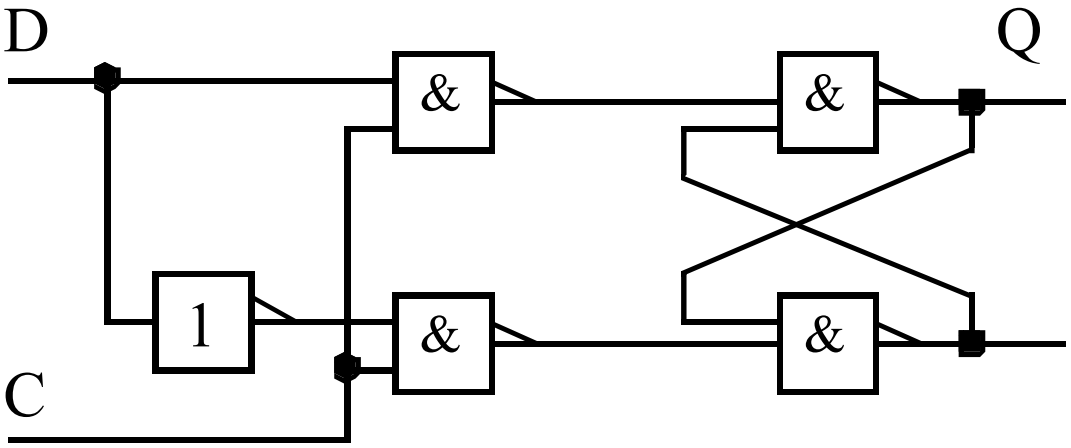
RAM Cell in RAS



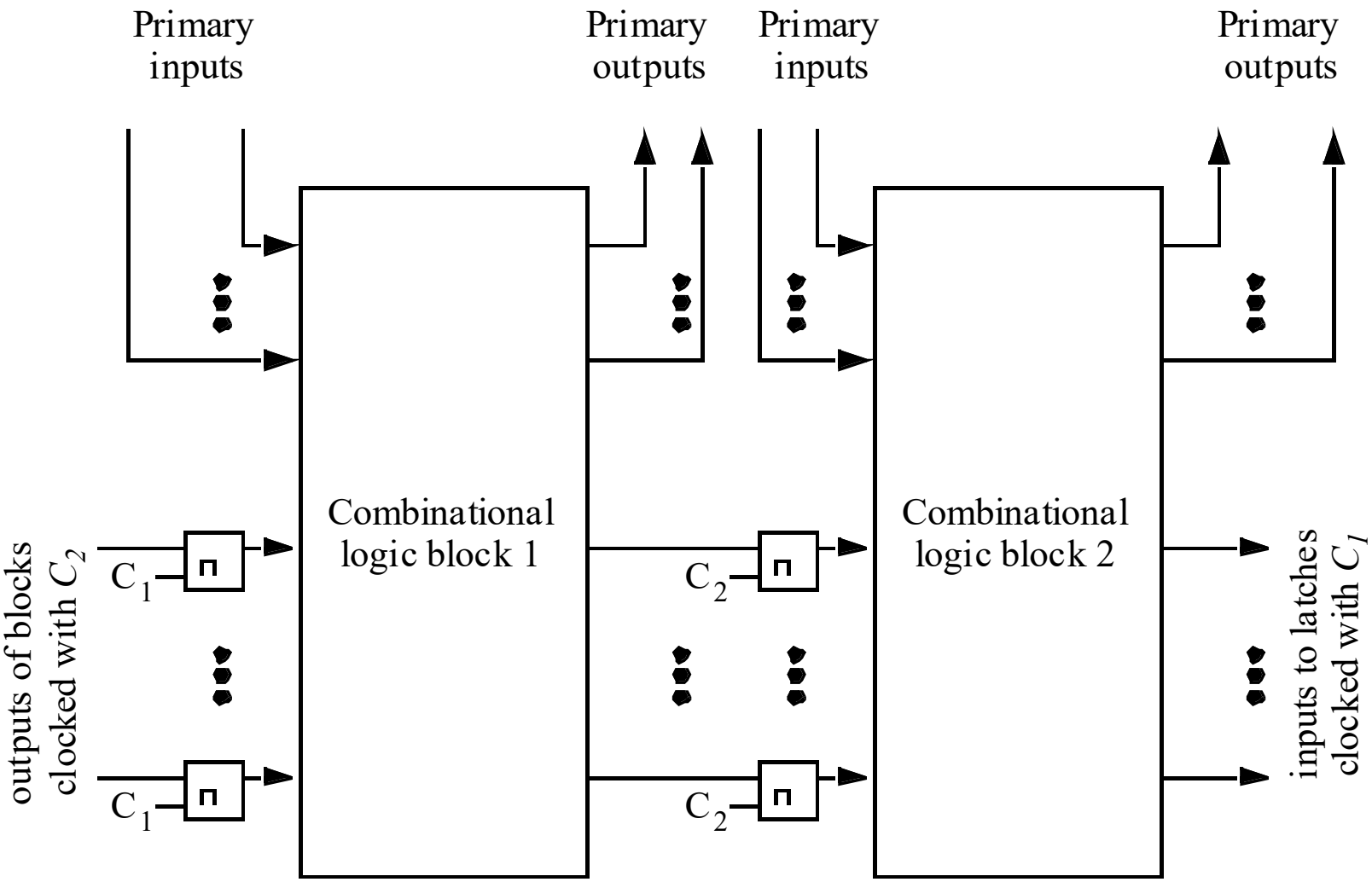
Single Latch Design - Cells

- Holds output and state constant when clock is low
- When clock is high, changes in *D* are reflected at its state as well as outputs with some delay
- Symbol and one possible implementation
- Used in certain types of circuits, e.g., high speed data paths



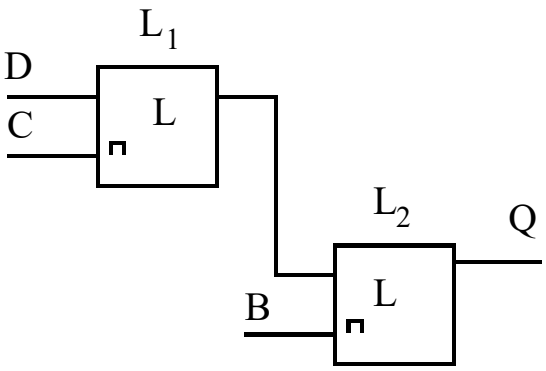


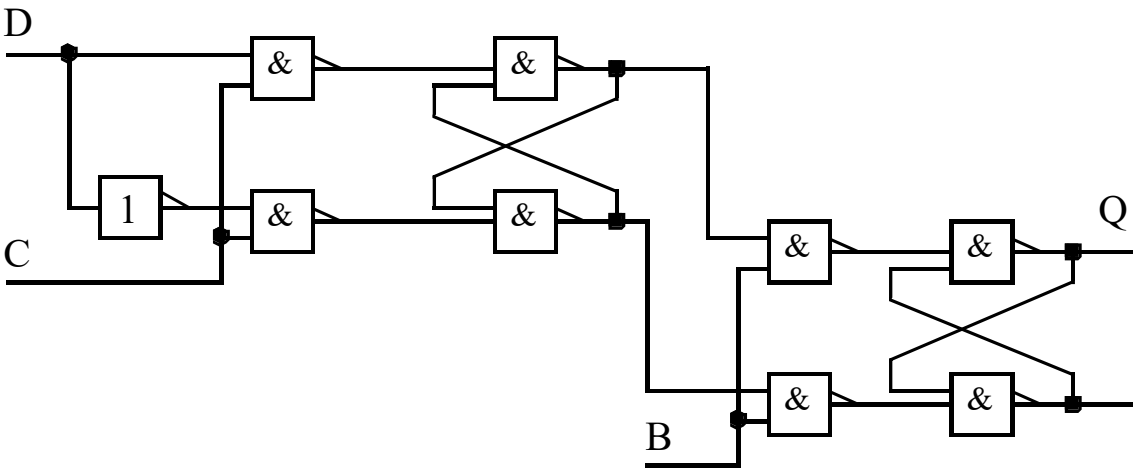
Single Latch Design - Configuration



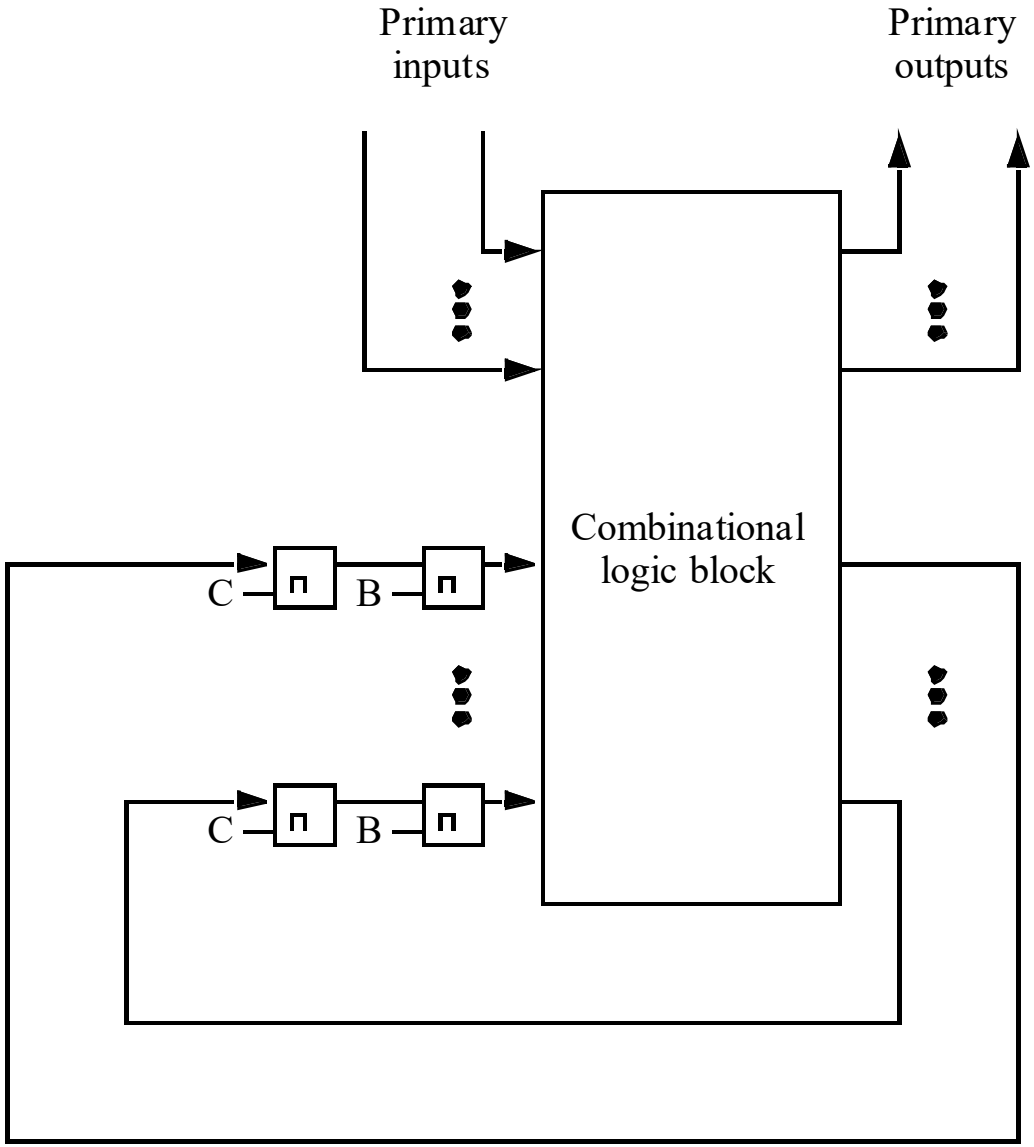
Double Latch Design - Cells

- Two clocks *B* and *C* must be non-overlapping, i.e., they must not assume the value 1 at the same time.
  - In particular, *B* can be complement of *C*



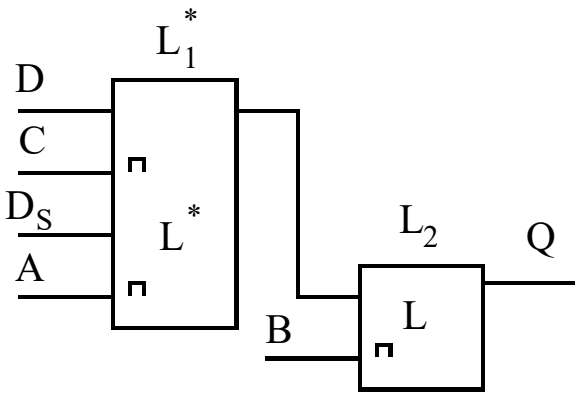


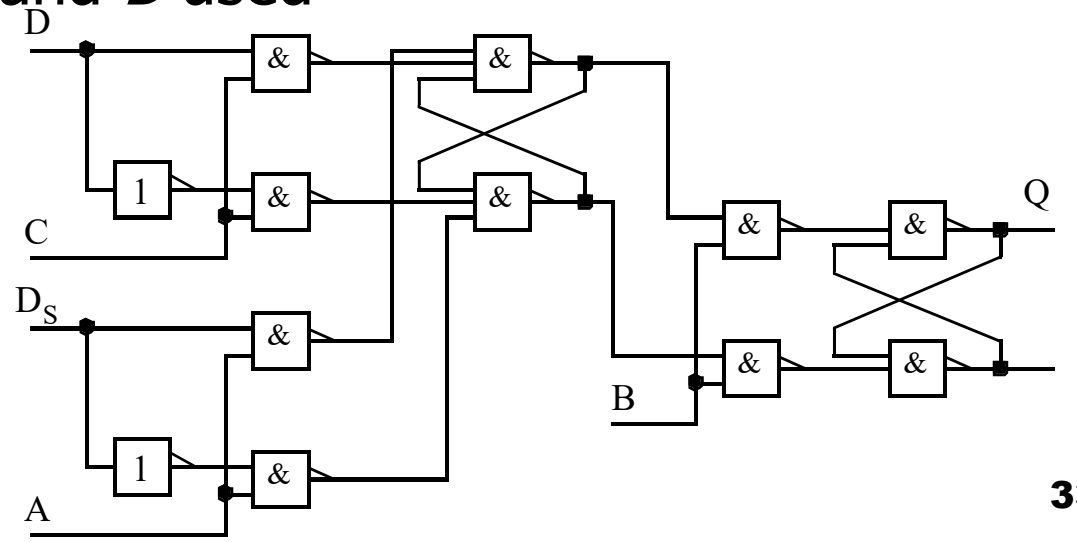
Double Latch Design - Configuration



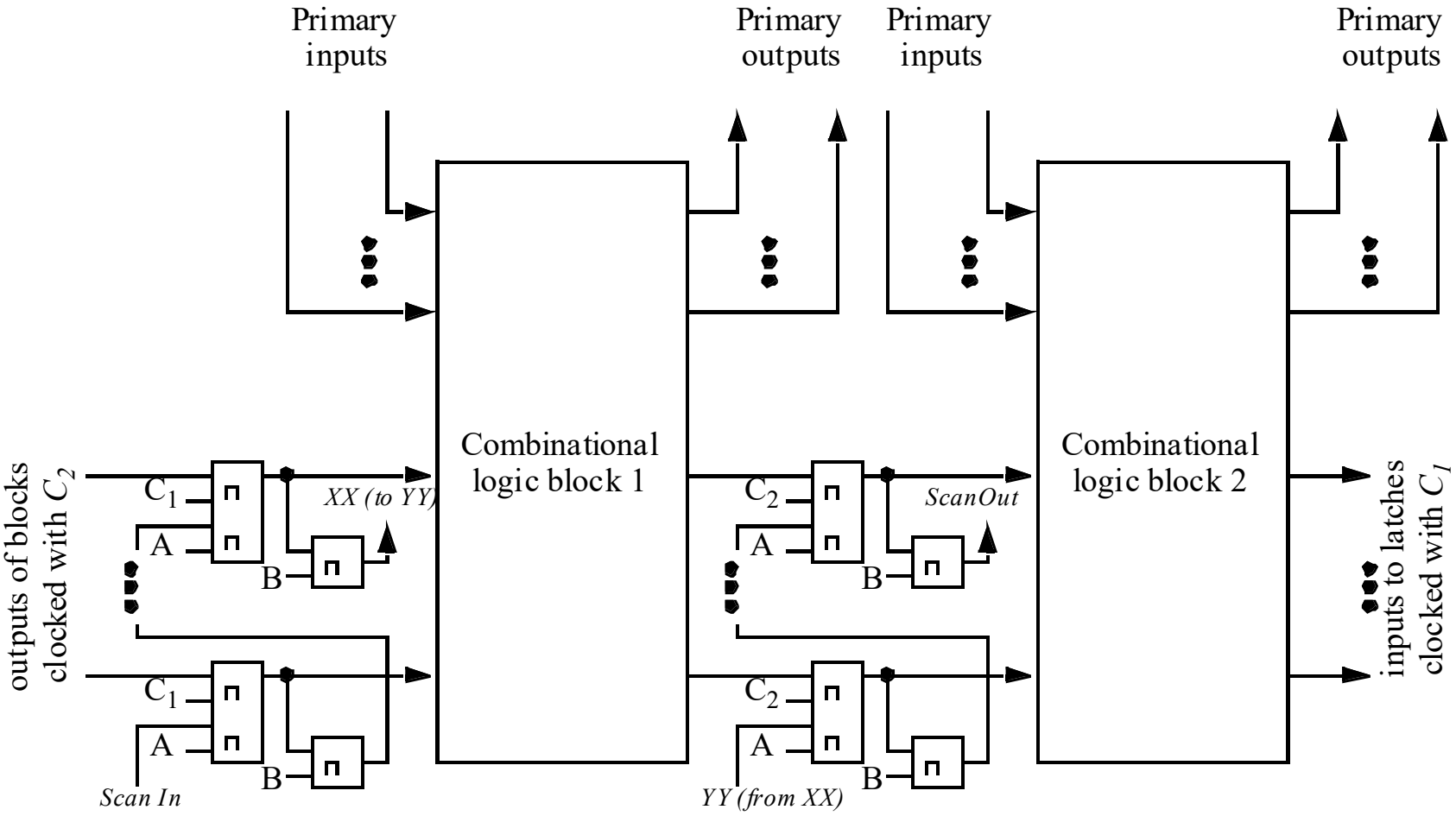
Level Sensitive (LS) Scan Element

- Latch L_1 in a standard master-slave configuration replaced by a two-port latch L^*, called L^*_1
- L^*_1 has
  - Two data inputs D and D_S
  - Two clocks C and A
- L_2 remains unchanged
- In normal mode, clock A held low (A=0) and non-overlapping clocks C and B used
- In test (scan) mode, clock C held low (C=0) and non-overlapping clocks A and B used

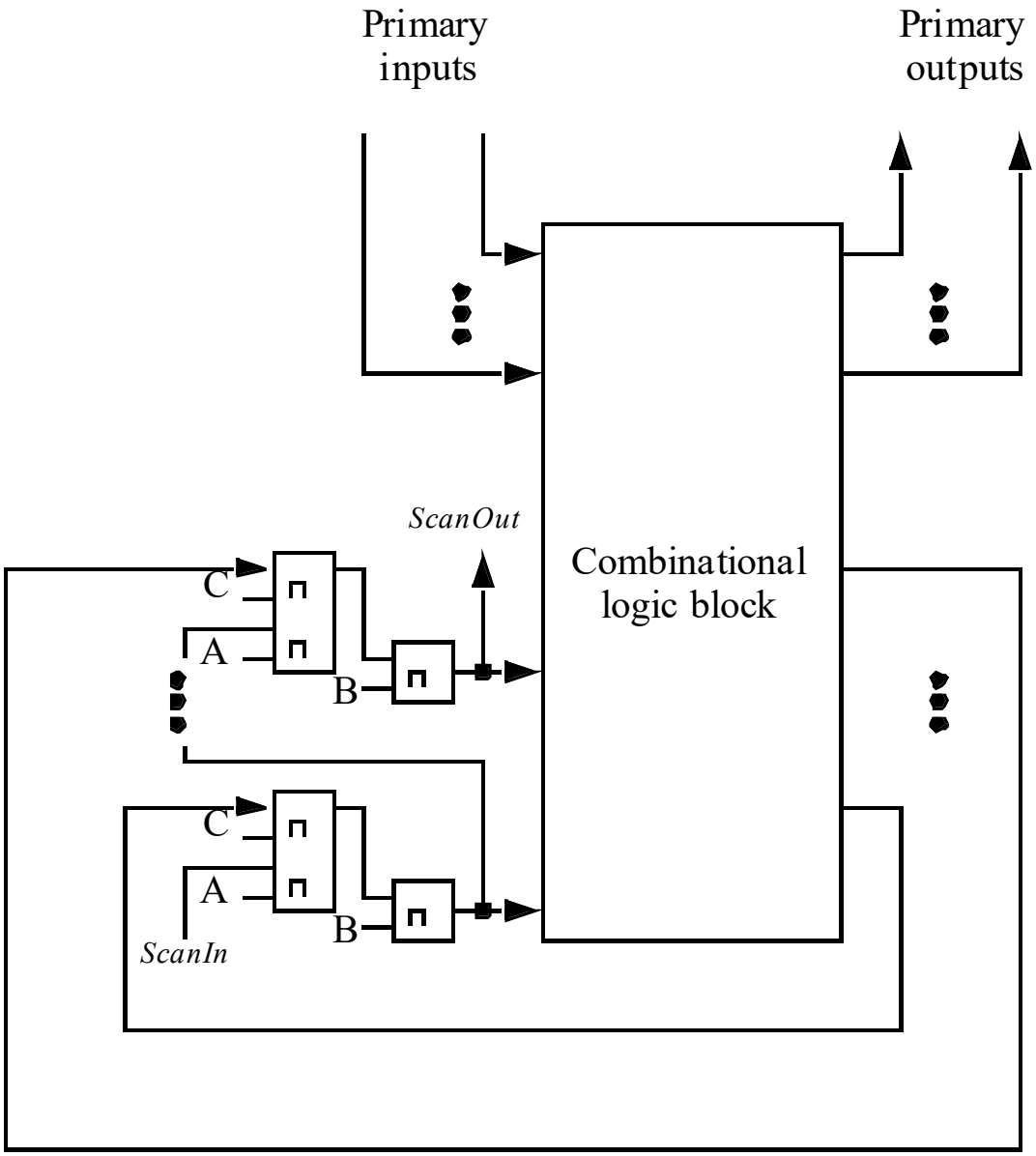




LS Element in Single Latch Configuration



LS Element in Double Latch Configuration



Scan Chain Organization

Organization of Scan Chains

- To organize scan flip-flops into chains by
  - —Partitioning them into multiple chains (if applicable)
  - —Ordering flip-flops within each chain
  - —Using reconfiguration circuitry (if applicable)
- To reduce test application time
- Model of circuit – assuming full-scan
  - —Gates and fanout systems combined into one or more maximal combinational blocks
    - Obtained by iteratively combining into one block any two circuit elements between which a combinational path exists
    - Called CLB_1, CLB_2, …, CLB_{Nb}

Organization of Scan Chains (cont.)

- Model of circuit – assuming full-scan
  - —All flip-flops whose normal data inputs are driven by outputs of CLB_i and whose outputs drive inputs of CLB_j (j may be equal to i) are combined into a register
    - Such a register is said to be a
      - + Receiver for CLB_i
      - + Driver for CLB_j
    - $Lreg_i$ is the length of register R_i, i.e., the number of flip-flops in R_i

Definitions & Metrics

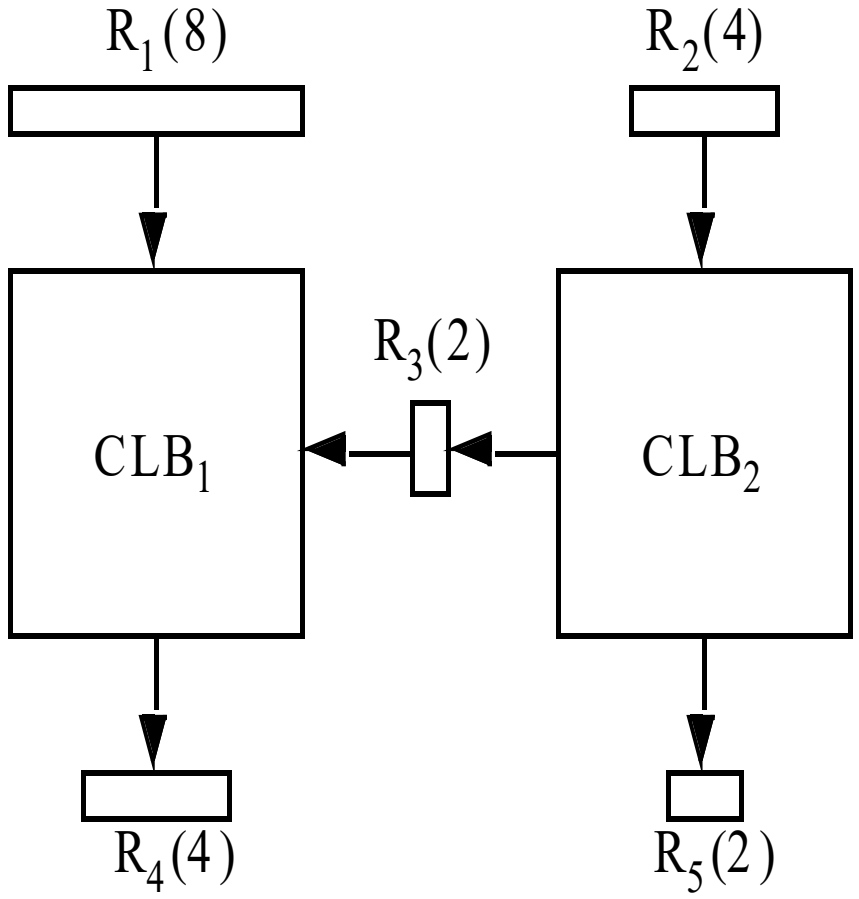
- A kernel is a sub-circuit that can be tested independently of the rest of the circuit
- In a full-scan circuit
  - A kernel (K_i) is synonymous with a maximal combinational block (CLB_i)
  - Number of kernels, $N_k = N_b$
- In a partial-scan circuit, a kernel may contain
  - One or more maximal combinational blocks
  - Non-scan registers

Definitions & Metrics (cont.)

- Register R_i is receiver for kernel K_j, if outputs of K_j drive normal data inputs of R_i
- R_i is driver for K_j, if outputs of R_i drive inputs of K_j
- R_i is a pure-driver if it is not a receiver for any kernel
- R_i is a pure-receiver if it is not a driver for any kernel

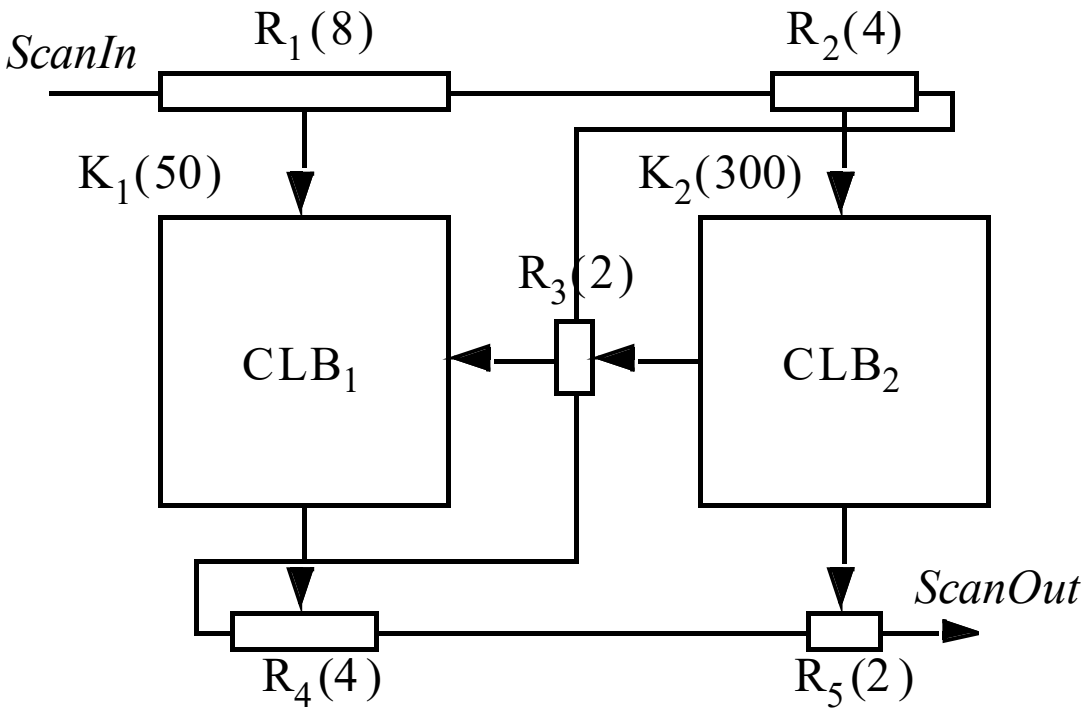
The Running Example

- R_1, R_2, R_3, R_4, R_5 registers of lengths 8, 4, 2, 4, and 2, respectively
- R_1 and R_2 pure drivers for CLB_1 and CLB_2, respectively
- R_4 and R_5 pure receivers for CLB_1 and CLB_2, respectively
- R_3 is a driver for CLB_1 and receiver for CLB_2



Organization Example – One Test Session

- Full-scan version of above example circuit
- K_1 and K_2 are identical to CLB_1 and CLB_2, respectively



Optimization Metrics

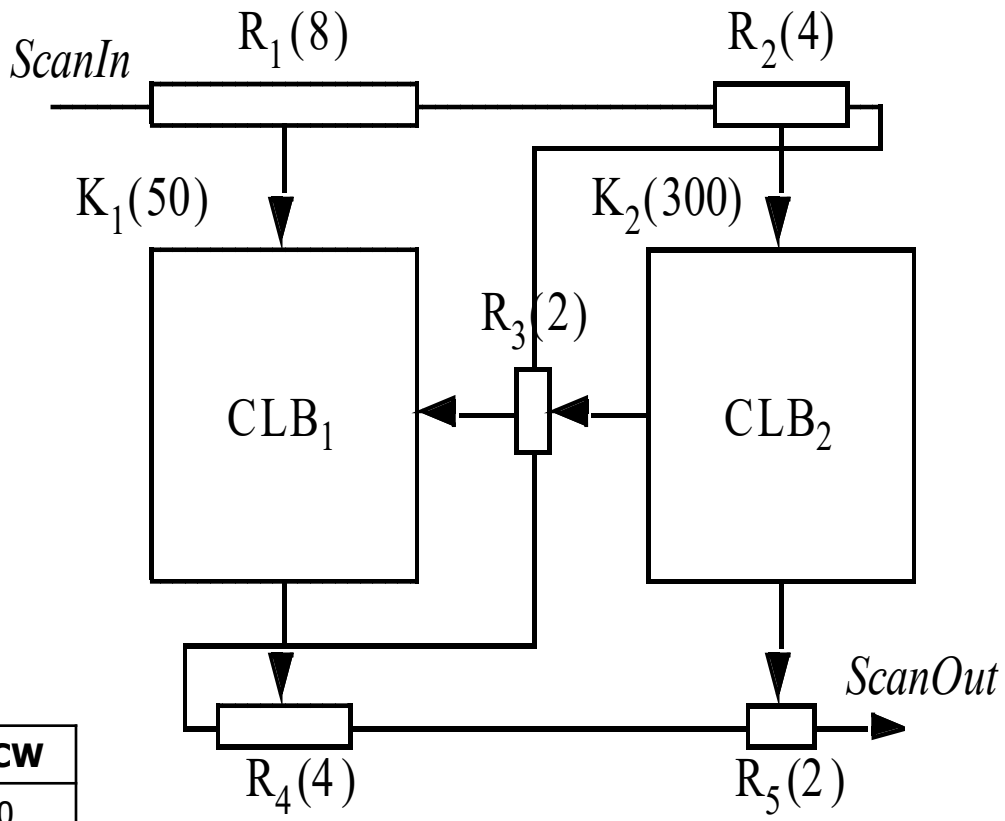
- Let NT_i be the number of test vectors for K_i
- Assume $NT_1 \cdot NT_2 \cdot \ldots \cdot NT_{Nk}$
- In a circuit with multiple (N_{sc}) scan chains
  - Chains are called $ScanChain_i$, $1 \cdot i \cdot N_{sc}$
  - $ScanChain_i$ has
    - Scan input output $ScanIn_i$ and $ScanOut_i$
    - SCL_i flip-flops
  - We assume synchronous scan, i.e., all chains share a single mode control (*mc*) signal

Optimization Metrics (cont.)

- Driving weight of a register R_i, DRW_i, is the number of vectors for which R_i is used as a driver
- If R_i is a driver for kernel K_j, then $DRW_i = NT_j$
- Receiving weight of a register R_i, RCW_i, is defined in a similar manner

Organization Example – Two Test Session

- *NT1 = 50* and *NT2 = 300*
- *R1* is pure driver with *DRW1 = 50* and, by definition, *RCW1 = 0*
- *R5* is pure receiver with *RCW5 = 300* and, by definition, *DRW5 = 0*
- *R3* is a driver/receiver with *DRW3 = 50* and *RCW3 = 300*



Register	Bitwidth	Type	DRW	RCW
R1	8	Pure Driver	50	0
R2	4	Pure Driver	300	0
R3	2	Driver/Receiver	50	300
R4	4	Pure Receiver	0	50
R5	2	Pure Receiver	0	300

Test Vectors vs. Test Sessions

- Characteristics of scan vectors
  - —Vectors may have many don't cares
  - —Different kernels require different numbers of vectors
- Desired values must be scanned into some registers for many more vectors than into others
- Responses captured in some registers need to be scanned out for many more vectors than those captured in others
- Test sessions
  - —A circuit may be tested in multiple test sessions
  - —In each test session, a different combination of kernels tested

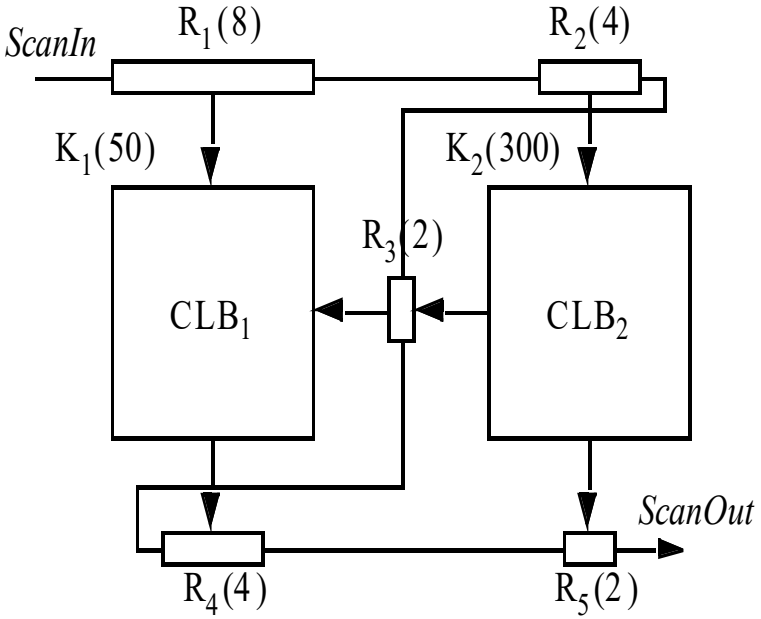
Test Application Schemes

- 1. Combined test application scheme
  - Circuit tested in one session
  - All kernels tested simultaneously
  - The number of vectors applied is the maximum of the number of vectors required for any kernel
- 2. Overlapped test application scheme
  - Assume $NT_1 < NT_2 < \ldots < NT_{Nk}$
  - Circuit tested in N_k sessions
    - In the first session, $NTS^{Ovr}(TSes_1) = NT_1$ vectors applied to all kernels
    - In the second session, $NTS^{Ovr}(TSes_2) = NT_2 - NT_1$ vectors applied to all kernels except K_1
    - In the third session, $NTS^{Ovr}(TSes_3) = NT_3 - NT_2$ vectors applied to all kernels except K_1 and K_2
    - …
    - In the last session, $NTS^{Ovr}(TSes_{Nk}) = NT_{Nk} - NT_{Nk-1}$ vectors applied to the last kernel

Test Application Schemes (cont.)

- 3. Isolated test application scheme
  - —In each session, one kernel is tested by itself
  - —In the first session, NT_1 vectors applied to K_1
  - —In the second session, NT_2 vectors applied to K_2
  - —…
  - —This mechanism is not of interest as it has no advantage over the other two.

Comparing Test Application Schemes

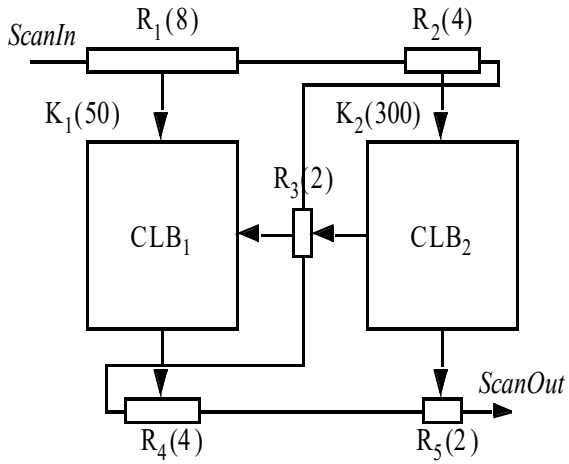


	$TSes_1$	$TSes_2$
Combined	300 vectors to K_1 & K_2	--
Overlapped	50 vectors to K_1 & K_2	250 vectors to K_2
Isolated	50 vectors to K_1	300 vectors to K_2

- Overlapped test application scheme most efficient

Role of Active Drivers/Receivers

- Active drivers in session $TSes_i$, $AD(TSes_i)$, is the set of drivers into which desired values need to be scanned in $TSes_i$
- Active receivers in session $TSes_i$, $AR(TSes_i)$, is the set of receivers from which responses need to be scanned out in $TSes_i$
- For the example circuit
  - $AD(TSes_1) = \{R_1, R_2, R_3\}$; $AR(TSes_1) = \{R_3, R_4, R_5\}$
  - $AD(TSes_2) = \{R_2\}$; $AR(TSes_2) = \{R_3, R_5\}$
- A chain is active in $TSes_i$ if any of the registers in the chain is active as a driver or receiver in $TSes_i$



Scan Shift/Flush Policy

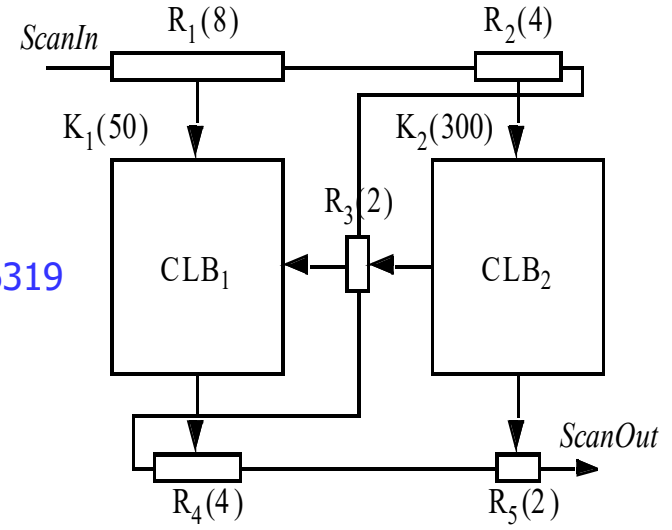
- Scan-shift policies
  - —Scan has two purposes
    - To scan out response captured for the previous vector
    - To scan in the next vector
- I. Flush policy
  - —For every chain
    - Content of every flip-flop scanned out
    - Test data scanned into every flip-flop
  - —If N_{sc} chains of lengths $SCL_1, SCL_2, \ldots, SCL_{Nsc}$
    - For application of each vector, chains configured in test mode, and
    - $SC^{Flsh} = max(SCL_1, SCL_2, \ldots, SCL_{Nsc})$ clocks applied
  - —SC^{Flsh} is called the shift cycle for the flush policy

Minimum Shift Policy

- II. Minimum shift policy – in a particular test session
  - —Shift test data into every active driver
  - —Shift response out of every active receiver
  - —If for $ScanChain_i$ the above two respectively require $SCin_i$ and $SCout_i$ cycles, then
    - $SC_i = max(SCin_i, SCout_i)$
  - —Hence, for that session, $SC^{MS} = max_i\, SC_i$

Example of Shift/Flush Policy

- For flush policy
  - $SC^{Flsh}(TSes_1)$ = 20 cycles (8+4+2+4+2)
  - $SC^{Flsh}(TSes_2)$ = 20 cycles
  - Time (Combined)=$n_{comb}(n_{sff}+1)+(n_{sff}-1)=300(20+1)+(20-1)=6319$
- Lower shift-cycle can be used (i.e. Minimum Shift)
  - In $TSes_1$
    - Test data needs to be scanned into R_1, R_2, and R_3
      - Hence, $SCin$ = 8+4+2 = 14
    - Response needs to be scanned out of R_3, R_4, and R_5
      - Hence, $SCout$ = 2+4+2= 8
    - Hence, $max(SCin, SCout)$ = 14 cycles sufficient for $TSes_1$
  - In $TSes_2$, only R_2 active as a driver and R_3 and R_5 as receivers
    - Hence, $SCin$ = 8+4 = 12, $SCout$ = 2+4+2 = 8
    - Hence, $max(SCin, SCout)$ = 12 cycles sufficient for $TSes_2$
  - Time (Overlapped) = [50(14+1)+250(12+1)]+(14-1) = 4013



Reg.	Bits	Type	DRW	RCW
R1	8	Dri	50	0
R2	4	Dri	300	0
R3	2	Dri/Rec	50	300
R4	4	Rec	0	50
R5	2	Rec	0	300

Active Shift/Flush Policy

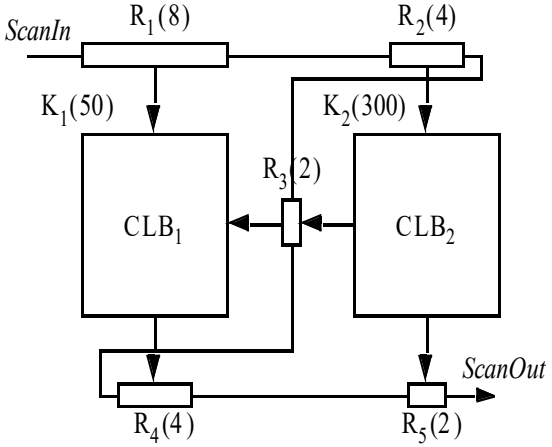
- III. Active flush policy
  - Flush policy applied only to active scan chains
  - For **single chain** designs, identical to flush policy
- IV. Active minimum-shift policy
  - Minimum shift policy applied only to active scan chains
  - For **single chain** designs, identical to minimum-shift policy

Summary of Scan Application/Shift

Number of Chains	Scan-Shift Policy	Test Application Schemes		
		Combined (Test all kernels in one session; $N_{vectors} = Max\{N_k\}$)	Overlapped (Test K kernels in K sessions; $N_{vectors} = N_1, N_2 - N_1, \ldots, N_k$)	Isolated (Test one kernel per session)
One	Flush (scan in/out into/from **all** FFs)			Has no advantage over other schemes/ policies. It's not of interest.
	Minimum-Shift (scan in/out into/from **active** driver/receiver FFs)		Minimum Overall Time	
Multiple	Active Flush (apply flush to **active** scan chains)			
	Active Minimum-Shift (apply minimum-shift to **active** scan chains)		Minimum Overall Time	

Comparison - Single Scan Chain Example

Reg.	Bits	Type	DRW	RCW
R1	8	Dri	50	0
R2	4	Dri	300	0
R3	2	Dri/Rec	50	300
R4	4	Rec	0	50
R5	2	Rec	0	300

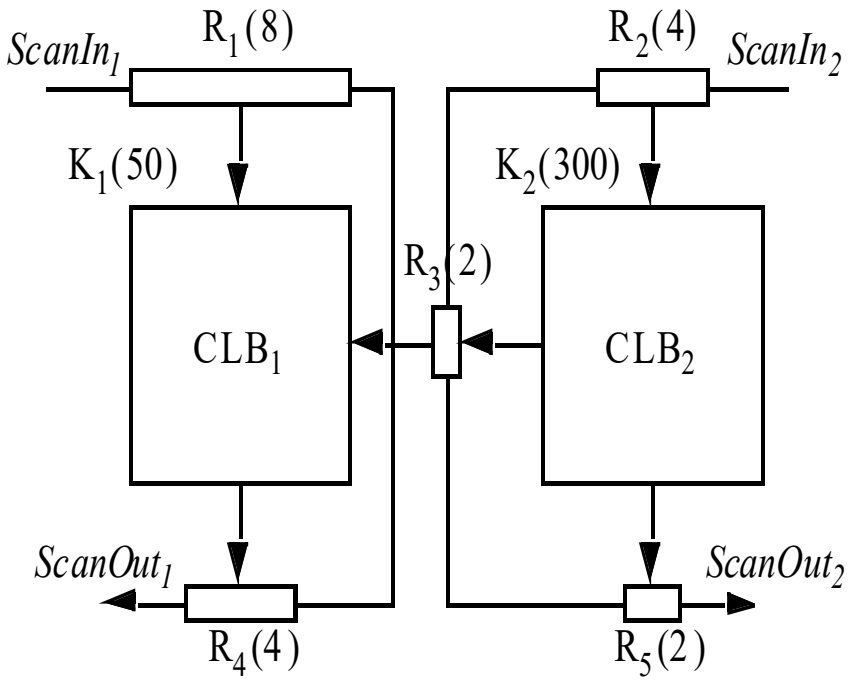


	Combined	Overlapped
Flush	6319	6319
Minimum-Shift	4513	4013

- Shift Policy: Flush; Application Scheme: Combined
  - Time = $[n_{comb}(n_{sff}+1)]+(n_{sff}-1) = [300(20+1)]+(20-1) = 6319$
- Shift Policy: Flush; Application Scheme: Overlapped
  - Time = [50(20+1)+250(20+1)]+(20-1) = 6319
- Shift Policy: Minimum-Shift; Application Scheme: Combined
  - Time = [300(14+1)]+(14-1) = 4513
    session 1: max input/output scan shifts = max{14,8} = 14
- Shift Policy: Minimum Shift; Application Scheme: Overlapped
  - Time = [50(14+1)+250(12+1)]+(14-1) = 4013
    session 1: max input/output scan shifts (CLB_1 and CLB_2) = max{14,8} = 14
    session 2: max input/output scan shifts (CLB_2) = max{12,8} = 12

Multiple Scan Chains

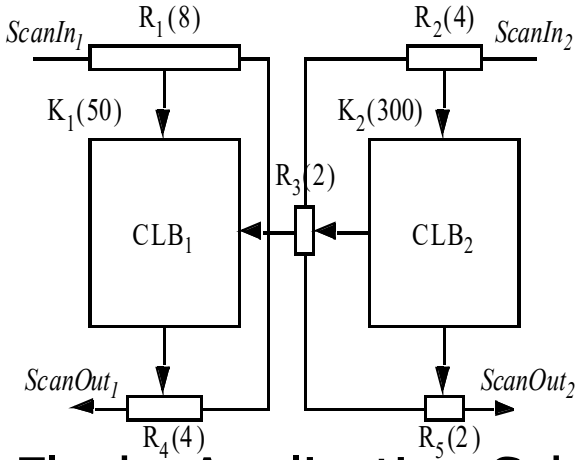
- This design has two chains
  - $ScanChain_1$ with $SCL_1 = 12$
  - $ScanChain_2$ with $SCL_2 = 8$
- Under overlapped test application
  - For active flush policy
    - $SC^{AFlush}(TSes_1) = \max\{12,8\} = 12$
    - $SC^{AFlush}(TSes_2) = 8$
  - For active minimum-shift policy
    - $SC^{AMS}(TSes_1) = 8$ and $SC^{AMS}(TSes_2) = 4$



Register	Bitwidth	Type	DRW	RCW
R1	8	Pure Driver	50	0
R2	4	Pure Driver	300	0
R3	2	Driver/Receiver	50	300
R4	4	Pure Receiver	0	50
R5	2	Pure Receiver	0	300

Comparison - Multiple Scan Chain Example

Reg.	Bits	Type	DRW	RCW
R1	8	Dri	50	0
R2	4	Dri	300	0
R3	2	Dri/Rec	50	300
R4	4	Rec	0	50
R5	2	Rec	0	300



	Combined	Overlapped
Flush	3911	2911
Minimum-Shift	2707	1707

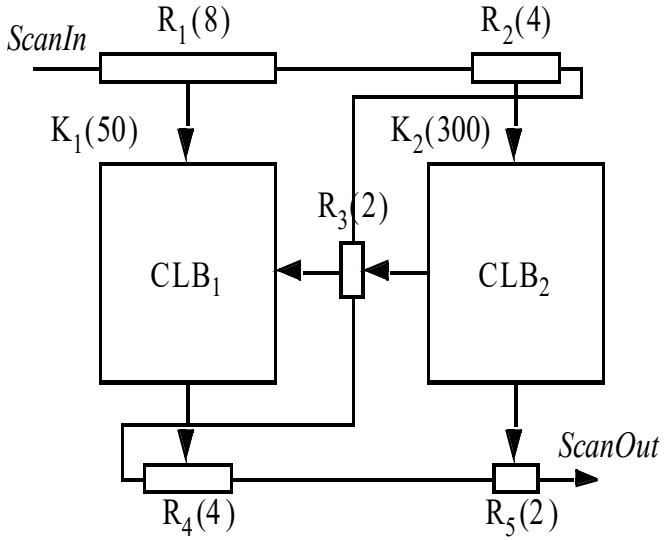
- Shift Policy: Active Flush; Application Scheme: Combined
  - Time =[300(12+1)]+(12-1)=3911 (max{12,8}=12 is max length of active chains)
- Shift Policy: Active Flush; Application Scheme: Overlapped
  - Time= [50(12+1)+250(8+1)]+(12-1)=2911
- Shift Policy: Active Minimum-Shift; Application Scheme: Combined
  - Time= [300(8+1)]+(8-1)=2707
    session 1: max active scan shifts (ScanIn1 and ScanIn2)=max{max{8,4},max{6,4}}=8)
- Shift Policy: Active Minimum Shift; Application Scheme: Overlapped
  - Time= [50(8+1)+250(4+1)]+(8-1)=1707
    session 1: max active scan shifts (ScanIn1 and ScanIn2)=max{max{8,4},max{6,4}}=8
    session 2: max active scan shifts (ScanIn2) = max{4,4}=4

Register Organization

- Organization of registers in a single chain depends upon
  - —The test application scheme
  - —Shift policy
  - —The characteristics of design, including
    - The numbers of tests required for each kernel
    - Sizes and driving/receiving weights of registers
  - —The key operation is ordering of registers in the chain
- Ordering of registers
  - —No impact for flush policy
  - —But has impact for minimum-shift policy

Register Organization – Single Chain #1

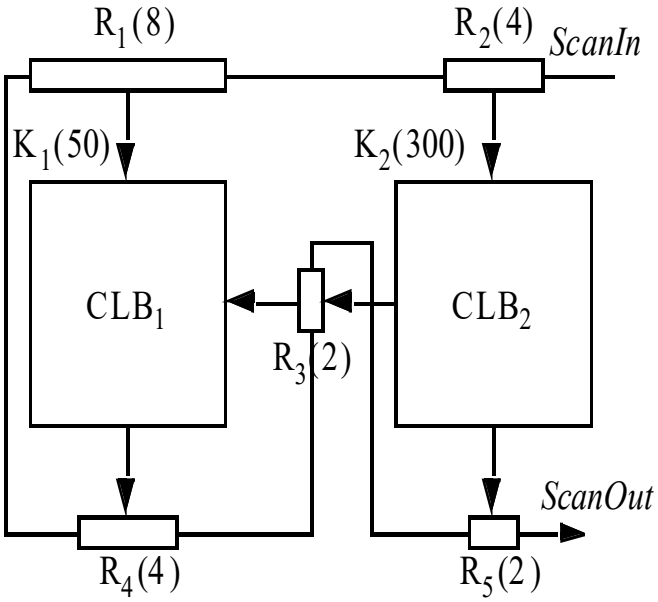
Reg.	Bits	Type	DRW	RCW
R1	8	Dri	50	0
R2	4	Dri	300	0
R3	2	Dri/Rec	50	300
R4	4	Rec	0	50
R5	2	Rec	0	300



- Organization: overlapped test application and minimum-shift policy
  - In TSes1, 50 vectors applied to CLB1 and CLB2 (Kernels K1 and K2)
  - In TSes2, 250 vectors applied to CLB2 (Kernel K2)
  - AD(TSes1) = {R1, R2, R3}
  - AR(TSes1) = {R3, R4, R5}
  - AD(TSes2) = {R2}
  - AR(TSes2) = {R3, R5}
  - SC(TSes1) = max{14,8} = 14 and SC(TSes2) = max{12,8} = 12
  - Hence, test application time for this order of registers
    - Time = [50(14 + 1) + 250(12 + 1)] + (14 – 1) = 4013 clock cycles

Register Organization – Single Chain #2

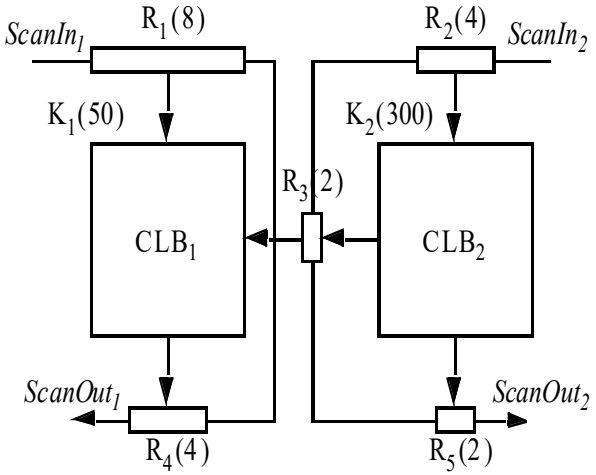
Reg.	Bits	Type	DRW	RCW
R1	8	Dri	50	0
R2	4	Dri	300	0
R3	2	Dri/Rec	50	300
R4	4	Rec	0	50
R5	2	Rec	0	300



- Organization: overlapped test application and minimum-shift policy
  - In TSes1, 50 vectors applied to CLB1 and CLB2
  - In TSes2, 250 vectors applied to CLB2
  - SC(TSes1) = max{18,8} = 18 and SC(TSes2) = max{4,4} = 4
  - Hence, test application time for this order of registers
    - Time = [50(18 + 1) + 250(4 + 1)] + (18 – 1) = 2217 clock cycles

Register Organization – Multiple Chains #1

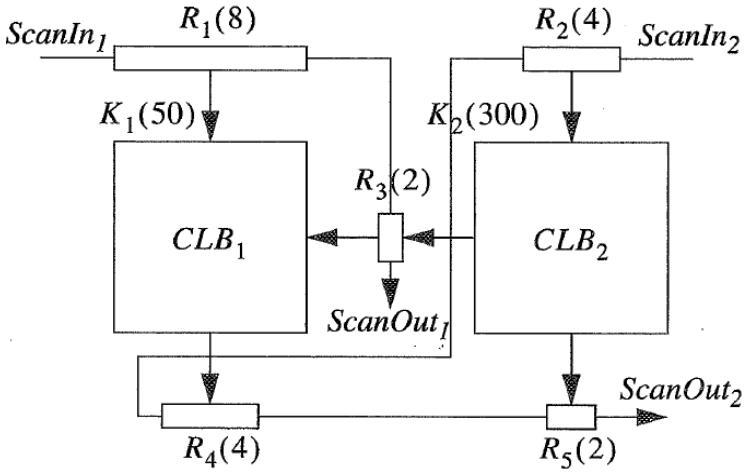
Reg.	Bits	Type	DRW	RCW
R1	8	Dri	50	0
R2	4	Dri	300	0
R3	2	Dri/Rec	50	300
R4	4	Rec	0	50
R5	2	Rec	0	300



- Organization: overlapped test application and minimum-shift policy
  - TSes1: 50 vectors applied to CLB1 and CLB2
  - TSes2: 250 vectors applied to CLB2
  - SC1(TSes1)=max{8,4}=8 and SC2(TSes1)=max{6,4}=6 → SC(TSes1)=max{8,6}=8
  - SC1(TSes2)= 0 (NOT ACTIVE), and SC2(TSes2)=max{4,4}=4 → SC(TSes2)=max{0,4}=4
  - Hence, test application time for this order of registers
    - Time= [50(8+1)+250(4+1)]+(8-1)=1707 clock cycles

Register Organization – Multiple Chains #2

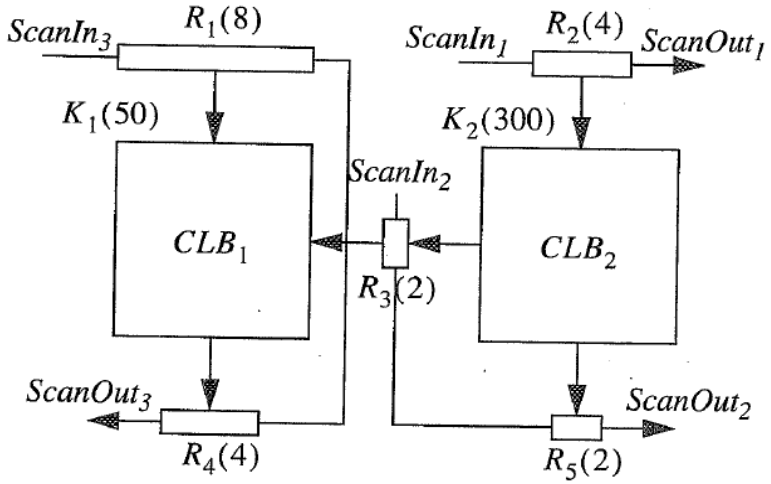
Reg.	Bits	Type	DRW	RCW
R1	8	Dri	50	0
R2	4	Dri	300	0
R3	2	Dri/Rec	50	300
R4	4	Rec	0	50
R5	2	Rec	0	300



- Organization: overlapped test application and minimum-shift policy
  - TSes1: 50 vectors applied to CLB1 and CLB2
  - TSes2: 250 vectors applied to CLB2
  - SC1(TSes1)=max{10,2}=10 and SC2(TSes1)=max{4,6}=6 → SC(TSes1)=max{10,6}=10
  - SC1(TSes2)= max{0,2}=2 and SC2(TSes2)=max{4,2}=4 → SC(TSes2)=4
  - Hence, test application time for this order of registers
    - Time= [50(10+1)+250(4+1)]+(10-1)=1809 clock cycles

Register Organization – Multiple Chains #3

Reg.	Bits	Type	DRW	RCW
R1	8	Dri	50	0
R2	4	Dri	300	0
R3	2	Dri/Rec	50	300
R4	4	Rec	0	50
R5	2	Rec	0	300



- Organization: overlapped test application and minimum-shift policy
  - TSes1: 50 vectors applied to CLB1 and CLB2
  - TSes2: 250 vectors applied to CLB2
  - SC1(TSes1)=max{4,0}=4, SC2(TSes1)=max{2,4}=4 and SC3(TSes1)=max{8,4}=8 → SC(TSes1)=max{4,4,8}=8
  - SC1(TSes2)= max{4,0}=4, SC2(TSes2)=max{0,4}=4 and SC3(TSes2)=0 (NOT ACTIVE) → SC(TSes2)=max{4,4,0}=4
  - Hence, test application time for this order of registers
    - Time= [50(8+1)+250(4+1)]+(8-1)=1607 clock cycles

Importance of Registers Positions

- The shift-cycle depends on the position of the registers in the chain
  - SCin(TSesi): The distance from ScanIn of the farthest driver that is active is TSesi
  - SCout(TSesi): The distance from ScanOut of the farthest receiver that is active is TSesi
  - SCMS(TSesi) = max{SCin(TSesi), SCout(TSesi)}

Guidelines for Register Organization

- Some simple rules
  - Place drivers closer to ScanIn than receivers
  - Place drivers with higher driving weights closer to ScanIn than drivers with lower driving weights
  - Place receivers closer to ScanOut than drivers
  - Place receivers with higher receiving weights closer to ScanOut than receivers with lower receiving weights
- These are only guidelines and do not guarantee optimality.

Guidelines for Register Organization (cont.)

- Rules conflict if some registers are drivers as well as receivers
- Hence, in general, necessary to enumerate various orders of registers
- Complexity can be reduced by using interchange property
  - If any scan chain contains R_i closer to *ScanIn* than R_j, where $DRW_i <= DRW_j$ **and** $RCW_i >= RCW_j$, then interchange R_i and R_j, provided
    - R_i is adjacent to R_j in the chain, or
    - $LReg_i = LReg_j$

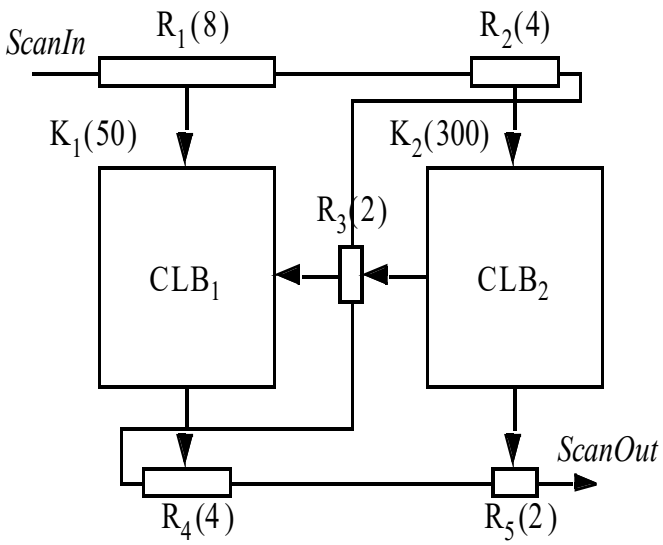
Special Cases

- Optimal ordering for special cases
  - —No driver-receivers
    - Order drivers in decreasing weights, followed by receivers in increasing weights
  - —All kernels require equal number of vectors
    - Pure drivers, followed by driver-receivers, followed by pure receivers
- For the general case, implicitly search all orders (utilize interchange property)

Guidelines for Multiple Scan Chains

- Organization of multiple scan chains
  - —Under the flush policy
    - Order of registers within a chain does not matter
    - The shift-cycle for any session is equal to the length of the longest chain
    - Hence, partition registers into multiple chains
      - Of equal lengths (if possible)
      - Or, if above not possible, then in such a manner that the length of the longest chain is minimized
  - —Under active flush methodology
    - Registers may be assigned in such a manner that one or more longer chains become inactive after the first few sessions

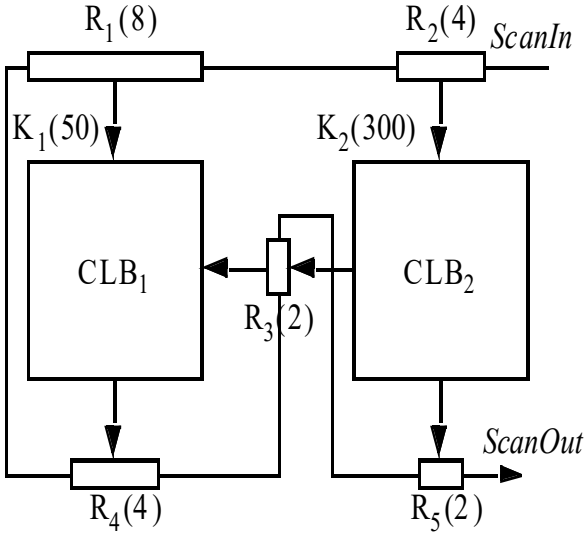
Example - Impact of Register Ordering



Better Order

- Pure Drivers
  - —DRW(R1)=50
  - —DRW(R2)=300
- Pure Receivers
  - —RCW(R4)=50
  - —RCW(R5)=300
- Driver & Receiver
  - —DRW(R3)=50
  - —RCW(R3)=300

Reg.	Bits	Type	DRW	RCW
R1	8	Dri	50	0
R2	4	Dri	300	0
R3	2	Dri/Rec	50	300
R4	4	Rec	0	50
R5	2	Rec	0	300

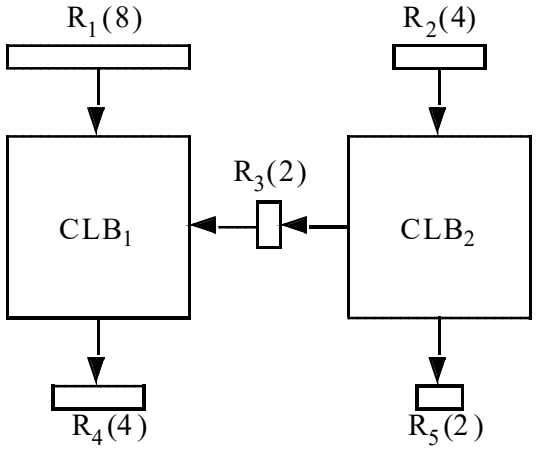


- Organization: overlapped test application and minimum-shift policy
  - — In TSes1, 50 vectors applied to CLB1 and CLB2 (Kernels K1 and K2)
  - — In TSes2, 250 vectors applied to CLB2 (Kernel K2)
  - — SC(TSes1) = max{14,8} = 14 and SC(TSes2) = max{12,8} = 12
- Hence, test application time for this order of registers
  - — Time = [50(14 + 1) + 250(12 + 1)] + (14 – 1) = 4013 clock cycles

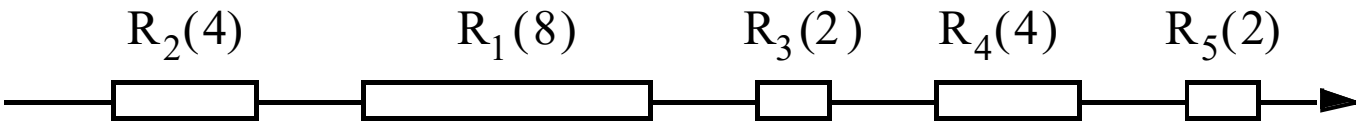
- Organization: overlapped test application and minimum-shift policy
  - — In TSes1, 50 vectors applied to CLB1 and CLB2
  - — In TSes2, 250 vectors applied to CLB2
  - — SC(TSes1) = max{18,8} = 18 and SC(TSes2) = max{4,4} = 4
- Hence, test application time for this order of registers
  - — Time = [50(18 + 1) + 250(4 + 1)] + (18 – 1) = 2217 clock cycles

Reconfigurable Chain

Reg.	Bits	Type	DRW	RCW
R1	8	Dri	50	0
R2	4	Dri	300	0
R3	2	Dri/Rec	50	300
R4	4	Rec	0	50
R5	2	Rec	0	300



- The chains can be reconfigured to bypass inactive registers
  - This chain minimizes the shift cycle for *TSes*$_1$



  - R_4 is inactive in *TSes*$_2$ but prevents minimization of shift cycle in *TSes*$_2$
  - This reconfigurable chain minimizes shift cycle for every session