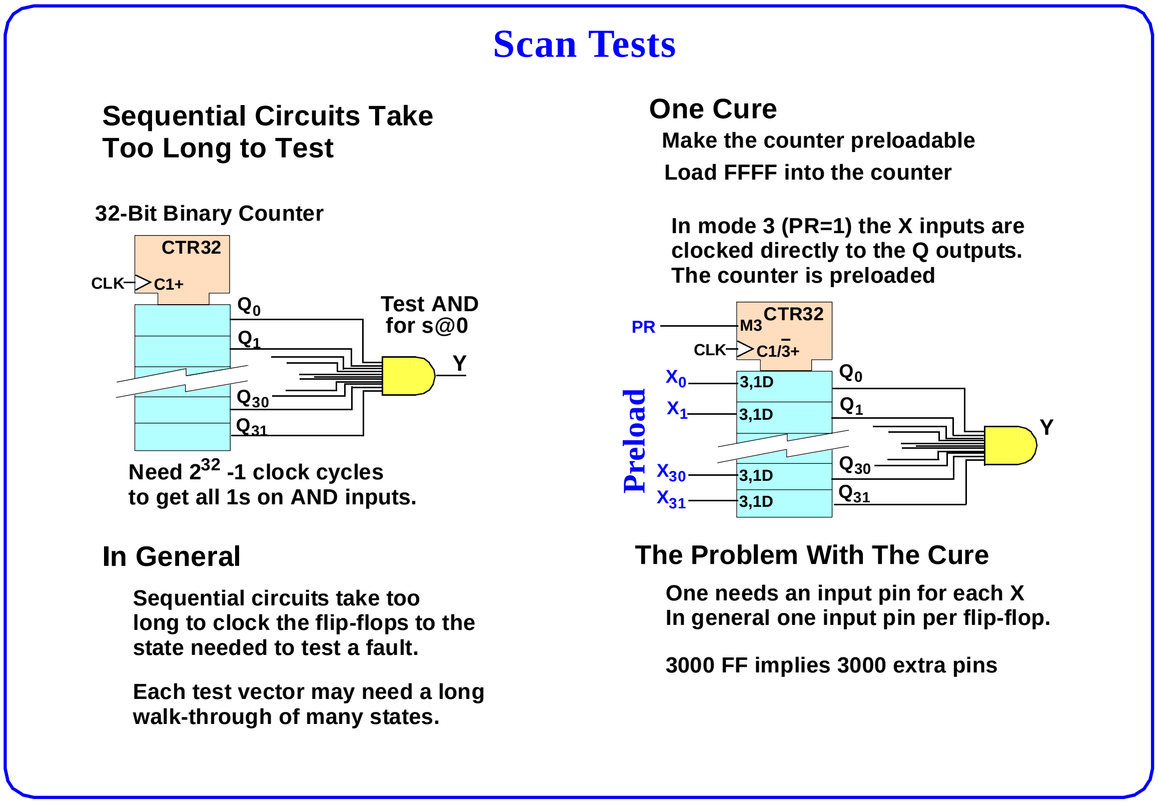


Scan Tests

Sequential Circuits Take Too Long to Test

32-Bit Binary Counter

Need 2^{32} -1 clock cycles to get all 1s on AND inputs.

In General

Sequential circuits take too long to clock the flip-flops to the state needed to test a fault.

Each test vector may need a long walk-through of many states.

One Cure

Make the counter preloadable

Load FFFF into the counter

In mode 3 (PR=1) the X inputs are clocked directly to the Q outputs. The counter is preloaded

The Problem With The Cure

One needs an input pin for each X
In general one input pin per flip-flop.

3000 FF implies 3000 extra pins

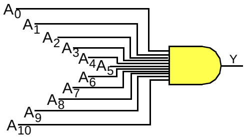
Scan Tests

The Sequential Circuit Testing Problem

To test combinational circuits

One uses a test vector which will give the wrong output if the circuit has a fault.

To test for Y, or any A_i, s@0 one uses test input $A_0A_1A_2$ ...A_9A_{10} = 111...11



To test sequential circuits

One must supply the right inputs and set the flip-flops to the right values to perform the test.

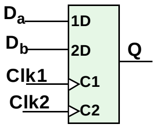
Setting the flip-flops may take many clock cycles. For the 32 bit counter it takes 2^{32}-1 cycles.

Symbols

The control signals in the inverted T ⊓ at the top of the counter are common to all the flip-flops.

PR is preload. A high PR sets the counter to mode 3 (M3), which is preload. The D inputs (D3,1) load the flip-flops when in mode 3. "3,1" means one needs both mode 3 and an active clock edge represented by "1".

A + on the clock input means a binary up counter.



The "1" on the clock input and some (or all) of the D inputs shows which other inputs are controlled by that clock.

A "1,$\overline{3}$+" on the clock input means the circuit is a counter from clk1 except in mode 3.

Sequential Circuit Testing Problem

A Better Cure

Make the preload in the form of a shift register.

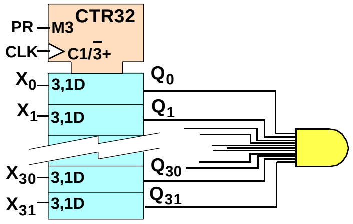
Only one extra pin, X_0, is needed to preload

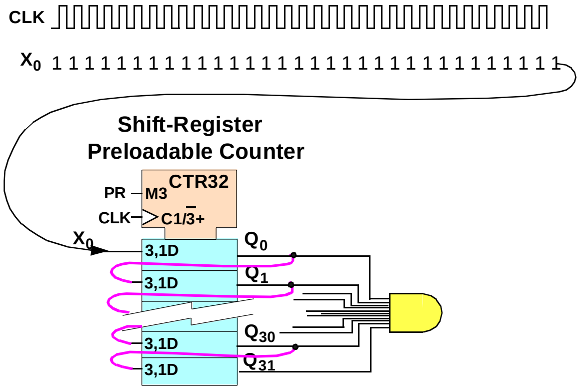
It takes only 32 clock cycles to preload (not good but acceptable)

Shift the desired 32 preload bits in through X_0
In this case 32 “1”s.

It takes 32 clock cycles instead of 2^{32}

Preloadable Counter





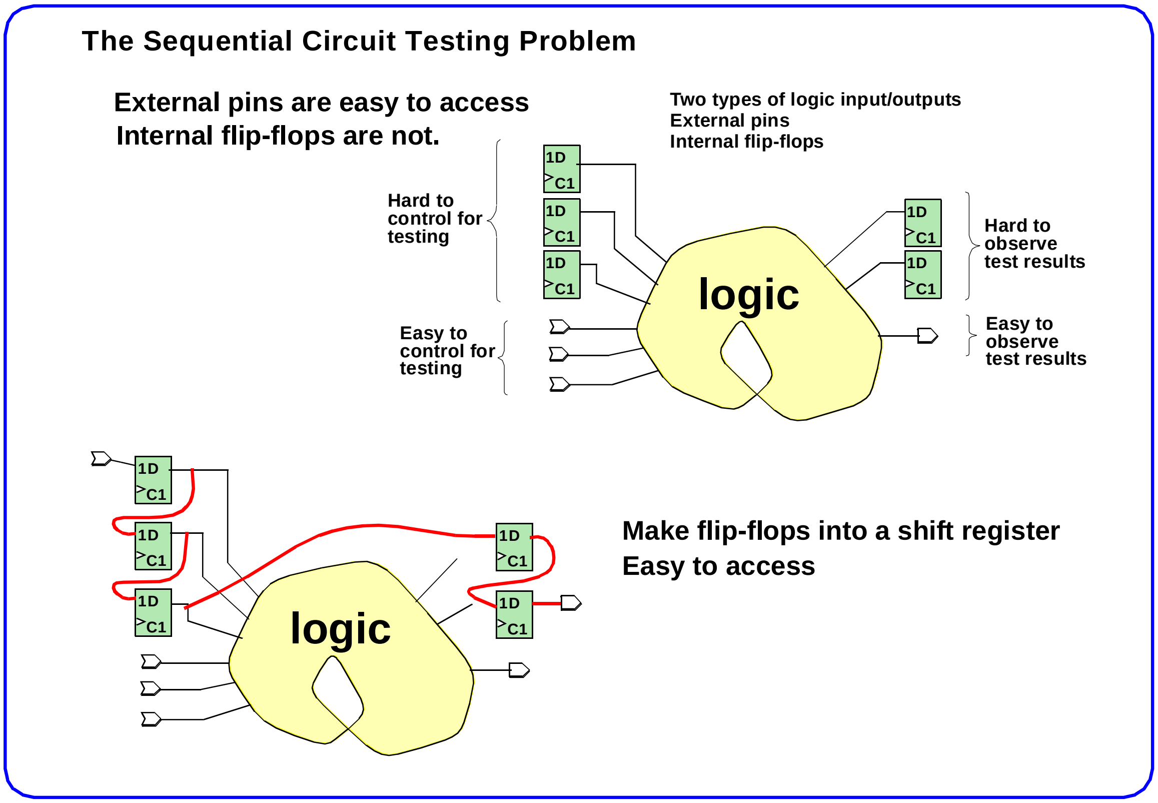
Shifting the test vectors in serially is the basis for scan testing

Scan Testing

Method

1. Convert the flip-flops into a shift register to load in a test vector.
2. Load the test vector by shifting the bits through the shift register.
   This takes N clock cycles. N is the number of flip flops.
3. When the vector is loaded, do the test.
   This takes one clock cycle.
4. Convert the flip flops back to a shift register.
5. Scan out the results of the test.
6. Go back to step 1 and using a new test vector.

Note that scanning in the new test vector is normally combined in the same shifting operation as scanning out the old vectors.



The Sequential Circuit Testing Problem

External pins are easy to access
Internal flip-flops are not.

Two types of logic input/outputs
External pins
Internal flip-flops

Make flip-flops into a shift register
Easy to access

Hard to control for testing
Easy to control for testing
Hard to observe test results
Easy to observe test results
1D
C1
logic
1D
C1
logic

Concept of Scan Test Test

Combinational logic has two types of inputs:

External pins
Outputs of internal flip-flops.

It may be vary hard to set the flip-flops to the desired values to test the combinational circuit. We say the flip-flops have poor controllability.

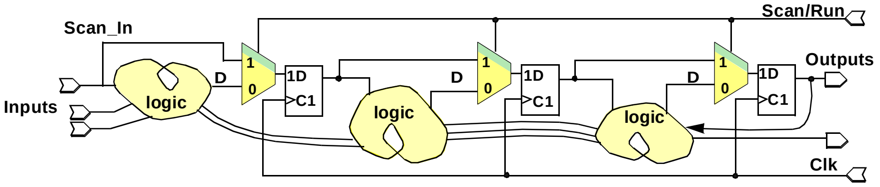
Combinational logic has two types of outputs:

External pins
Outputs which go to the D inputs of internal flip-flops.

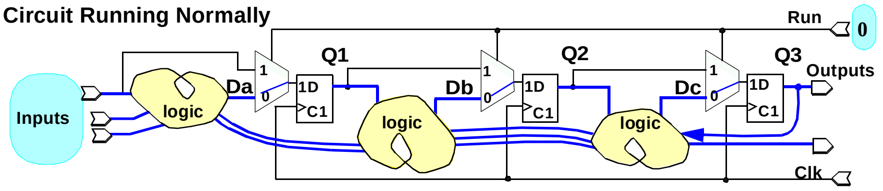
It is easy to look at the external pins. It may be vary hard to transfer the contents of the flip-flops to external pins. We say the flip-flops have poor observability.

Basic Scan Testable Circuit

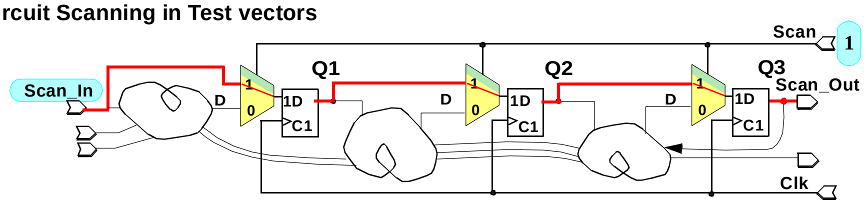
Scan/Run = 0; Flip-Flops part of normal circuit
= 1; Load test vector into flip-flops



Circuit Running Normally



Circuit Scanning in Test vectors



Basic Scan Test

Test Muxs Added to the Circuit

The original circuit has muxes added in from tof each flip-flop for testing.

The circuit has one to three extra I/O pins

a. Scan/Run wich changes the sircuit from scan-test mode to normal operation mode.

b. Scan_in which is a serial input for test vectors. This may be used as a normal input in Run mode.

c. Scan_out which is a serial output for the test results. This may be used as a normal output during Run mode.

Normal operation

Set Scan/Run to Run.
The muxs disconnect the test shift register.

Circuit becomes the original circuit.

Scan In/Outt

Set Scan/Run to Scan

Test vectors can be scanned into the flip-flops.

Outputs of the tests on the combinational logic can be shifted out.

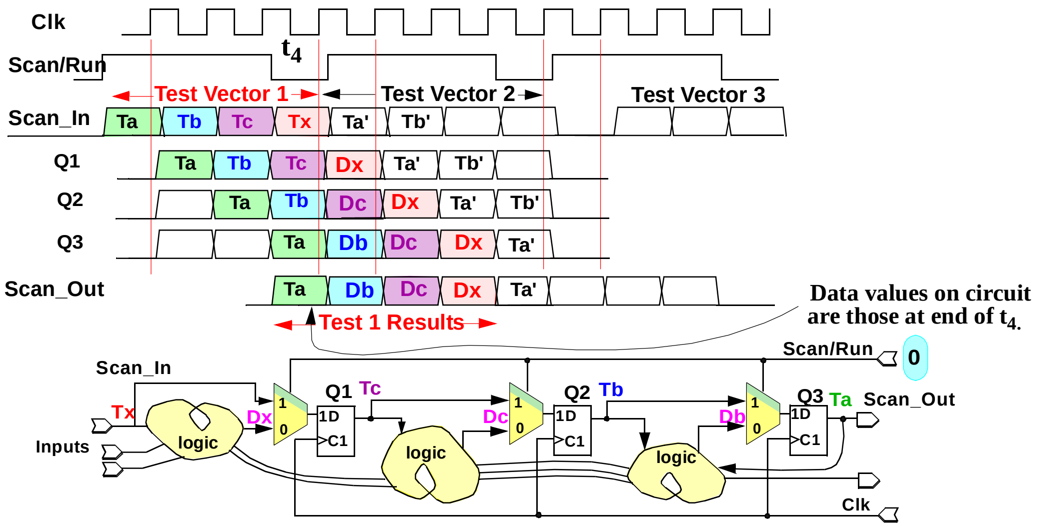
▪ Scan Tests ▪

Basic Scan Test

Method

1. Switch the muxs to SCAN
2. Shift in test vector. One clock cycles per FF.
3. Switch to muxs to RUN
   Clock logic outputs into flip-flops.

4. Switch the muxs to SCAN
5. Shift logic outputs Dx, Db, Dc thru FFs to Scan_Out.
   Simultaneously shift in new test vector.



Data values on circuit are those at end of t_4.

Basic Scan Test

Method

1. Switch the MUXs to Scan.
   This connects all the flip-flops as a shift register. It disconects the output of the logic blobs from reaching the flip-flop inputs.
2. Load the test vector. A test vector, designed to test the logic blocks, is shifted into the flip-flops. This takes as many clock cycles as there are flip-flops in the shift register.
3. Switch the MUXs to Run for one clock cycle
   This reconnects the logic blocks, and makes the circuit run normally for one clokc cycle.
   The outputs of the combinational logic blobs will appear as:
   a. outputs from the circuit
   b. inputs to the flip-flops (Dx, Dc and Db).
4. Clock the combinational logic results into the flip-flops.
5. Switch the MUXs to Scan
   At the same time you will clock out Ta, Db, Dc and Dx and they can be checked for errors.

The data values Tx, Dx, Tc, Dc, Tb, Db, and Ta are shown on the circuit in the positions they would be in just before the clock edge at the end of cycle t_4.

Scannable Flip-Flops

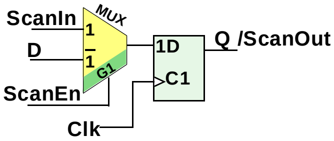
Types:

Multiplexed Flip-Flop

Flip-Flop with a MUX in front.

Most common.

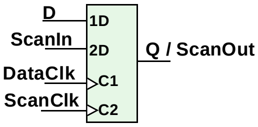
Mux may be buit into FF, or
Synthesizer can create from MUX and FF



Dual-Clock Flip-Flop

Special flip-flops with two clocks

ScanClk takes the place of *ScanEn*

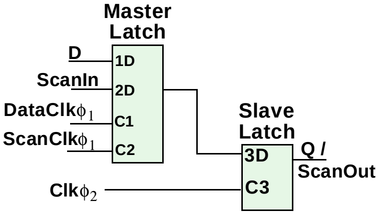


Level-Sensitive Scan Latches

Latches make Master-Slave FF

- With two clocks on master latch

Used by IBM



Scannable Flip-Flops

Multiplexed Flip-Flop

- The most common type.
- The synthesizer can construct it from a mux and a normal flip-flop if this cell is not in the library. However it is smaller, faster, and scan-safe (described later) if a special cell is designed.
- Only one noncritical global lead (scan/run) is needed.
- Only one extra pin is needed. Scan_in and scan_out can time-share with normal I/O pins.

Dual-Clock Flip-Flop

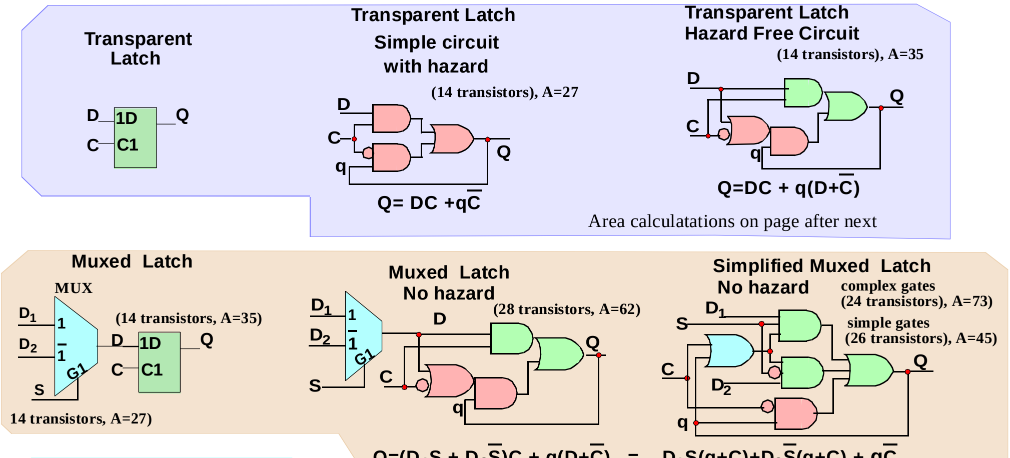
- The fastest scannable flip-flop. It’s setup time is only a few percent longer than a normal flip-flop.
- It requires two global leads. Both critical i.e. clocks.
- Best if scan is omitted for some flip-flops (partial scan).

Level-Sensitive Scan Latches

- Smallest increase in setup time
- Best for partial scan
- Must route three clocks.

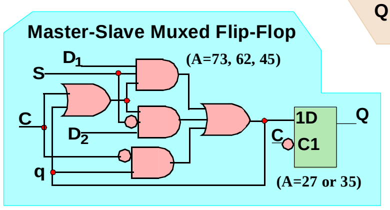
Mux Flip-Flop Circuits

Area Estimates



$$Q=(D_1S + D_2\overline{S})C + q(D+\overline{C}) \quad = \quad D_1S(q+C)+D_2\overline{S}(q+C) + q\overline{C}$$

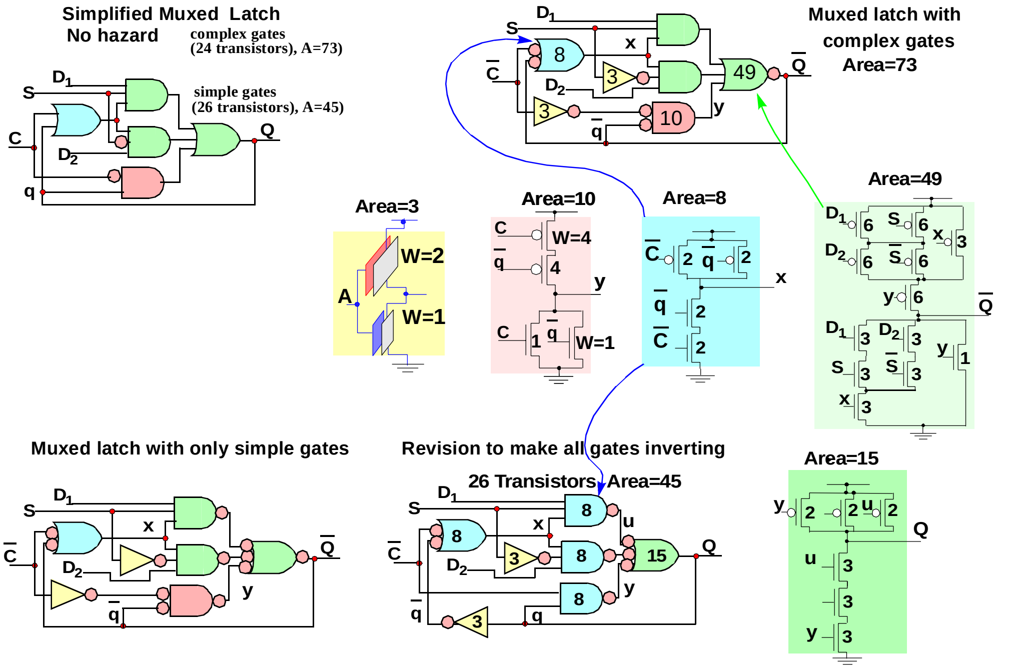
Area calculated on next page



Logic manipulation has reduced the path D1->Q to one complex gate, albeit one larger and slower than the original latch.The D->Q path and the load on Q determines the setup time which is the critical delay in fast designs. Setup time will increase and is certainly more than for the dual-clock FF, mainly because of the doubled load on Q.

The areas were estimated by sizing transistors to give the same total channel resistance in all gates. This gives a FF area increase of 40% to 100%

Details of Calculations for Area Estimates



▪ Scannable Flip-Flops ▪

Multiplexed Scannable Flip-Flops

Multiplexed Flip-Flop

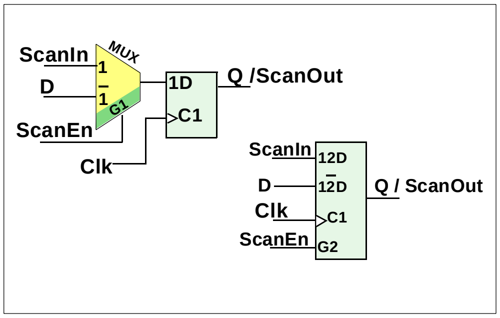
- The simplest scannable flip-flop.

Properties

1. Flip-flop area penalty: + 15% to 40%
   Circuit area penalty:[1]   + 5% to 15%.
2. Adds only one extra noncritical line.
   No extra clock line.
3. Speed penalty:
   MUX increases setup time 15 to 100%[2]
4. Best for flip-flops only
5. Can make from MUX if not in cell library.

   Synopsis test insertion can build them automatically.

   But resulting MUX may not be scansafe for EN, SET and RESET.

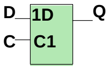


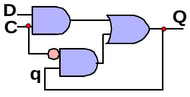
[1] Circuit area estimates are from Texas Instrument's, *Submicron ASIC Products Design for Testability*, 1995, Chapt 6.

[2] Synopsys Test Compiler Reference Manual 1992, p 3-4, says 40% to 75%

Area Estimates for D Latches

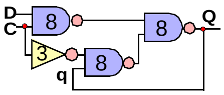
Transparent Latch





$Q=DC + q(\overline{C})$

(14 transistors), A=27



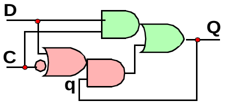
$Q=DC + q(\overline{C})$

The smallest D latch circuit.

A small, fast latch circuit

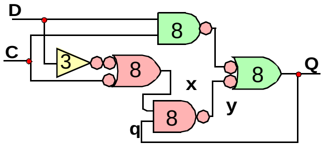
Unfortunately
$Q = DC + q(\overline{C})$
has a hazard in C
when D,q=1,1.
The circuit may be used if
analog simulations show
the glitch is absorbed.

Implementations of the larger hazard free circuit.



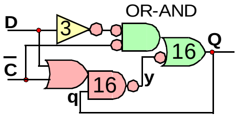
$Q=DC + q(D+\overline{C})$

(18 transistors), A=35

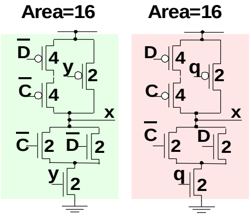


$Q=DC + q(D+\overline{C})$

(14 transistors), A=35



$Q=DC + q(D+\overline{C})$



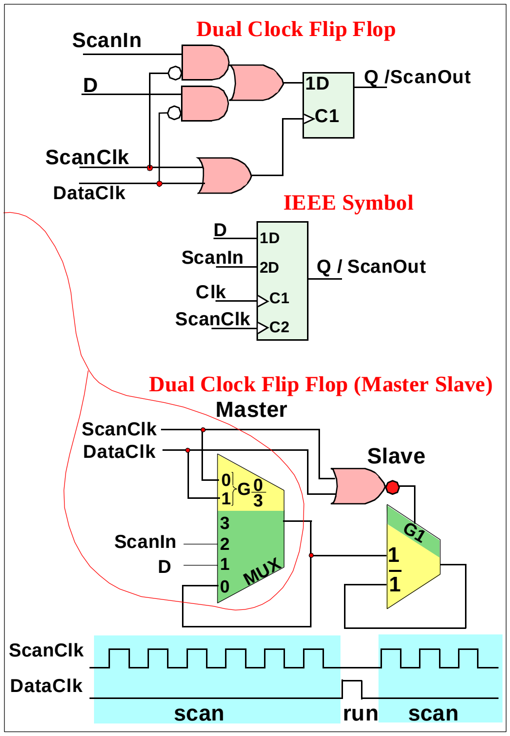
Clocked Scan Flip-Flops

Dual-Clock Flip-Flop

Properties

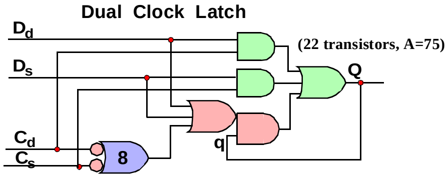
1. Small 5% increase in setup time. "MUX" can be merged with input latch.
2. Best for FF only. For testing, it will replace latches with edge-triggered FF.
3. Area penalty: FF area increases by 60% Circuit area increase typically 5% to 20%.
4. Adds an extra clock line, but no extra control line.
5. For partial scan: During scan in, the state of all nonscanned FF must be maintained.

- With MUXed scan must gate clock.
- Here just omit the *ScanClk* to the omitted flip-flops.

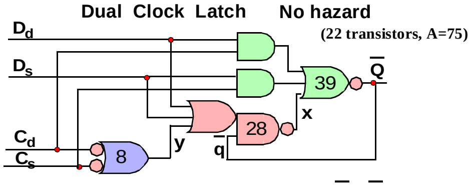


Dual-Clock Flip-Flop Circuits

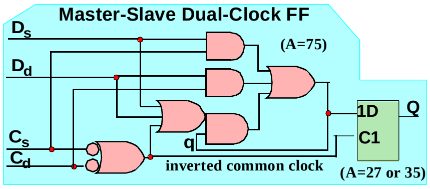
Rough Area and Delay Estimates



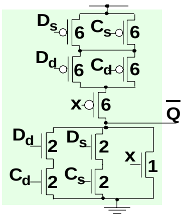
$Q=(D_sC_s + D_dC_d) + q(D_s+D_d+\overline{C}_s+\overline{C}_d)$



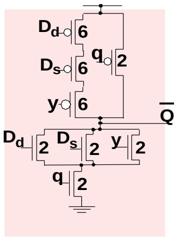
$Q=(D_sC_s + D_dC_d) + q(D_s+D_d+\overline{C}_s+\overline{C}_d)$



Area=39



Area=28



Logic manipulation again has reduced the path D->Q to one complex gate not much slower than in the original latch.The load on Q has not increased and the output gate has 3 parallel inputs instead of 2. Setup time will increase little, much less than for the mux FF.

The areas estimates used transistors sized to gave the same total channel resistance in all gates. This gives an area increase of 60% over the nonscanned flip-flop.

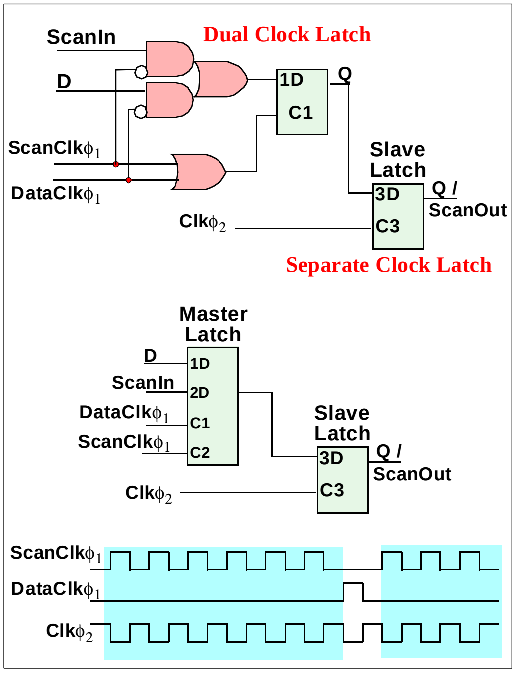
▪ Scannable Flip-Flops ▪

LSSD Scan

Level-Sensitive Scan Flip-Flop

Properties

1. Good for partial scan.
   - With MUXed scan must gate clock.
   - Here just omit the scanClk to the omitted flip-flops.
2. Level Sensitive has nothing to do with scan. It was what we would now call synchronous design, i.e:
   - No gating the clock.
   - No using asynchronous reset for logic, only on power up.
   - No RS latches, etc.
3. Area penalty:
   FF area increased by 15% to 30%
   Circuit area increase typically 4% to 20%.
4. Adds two extra clock lines.
5. Small 5% increase in setup time.

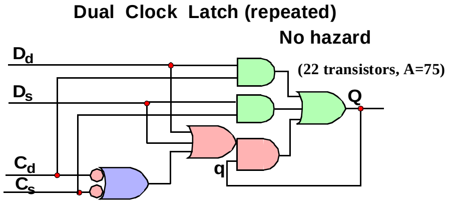


LSSD Flip-Flop Circuits

The input latch is a dual clock latch, the same as the input latch for the dual-clock flip flop.
The second (slave) latch runs from a separate clock.

Some firms, notably IBM, design with an LSSD methodology.
The main use for others, is for partial scan.
Having to route three clocks everywhere is a major drawback.

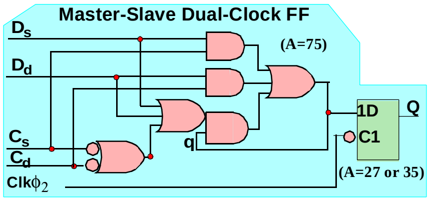
LSSD, Rough Area and Delay Estimates



$Q=(D_sC_s + D_dC_d) + q(D_s+D_d+\overline{C}_s+\overline{C}_d)$

The path D->Q is one complex gate not much slower than in the original latch. Setup time will increase little, but is much less than for the mux FF.

The areas were estimates used transistors sized to gave the same total channel resistance in all gates. This gives an area increase of about 60% over a nonscan flipflop.



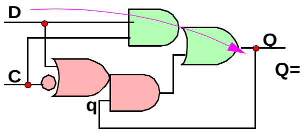
▪ Scannable Flip-Flops ▪

Hazard-Free Scan FF Comparison

Type of FF	FF Area	%incr	Tsetup* num gates	Tsetup Log effrt	Num Clks/ Num Enab
FF with no scan	70	0%	2	17	1/1
Mux + FF	97	38%	4	37	1/1
Mux combined with FF, one cell	80	14%	2	20	1/1
Dual-clock FF	110	57%	2	19.5	2/0
LSSD FF	110	57%	2	19.5	3/0

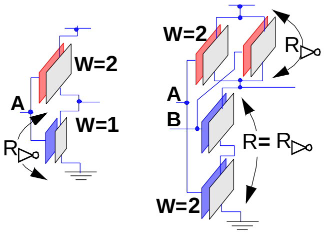
Area is ΣW taking L=1.
W taken to make all series resistances equal to R▷o.
Equalizes gate delays for small loads.

Tsetup* is the number of gates from D to Q-q loop.



$Q=DC+q(D+\overline{C})$

Other Tsetup is calculated by logical effort.



Res. of inverter channel=R▷o
W = channel width
L = channel length=1

Comparing Scannable Flip Flops

Area Comparison

The Basic flip flop circuits were optimized and the transistors sized to make the total channel resistance from output to ground, and output to V_{DD}, the same for all gates.

The flip-flops were constructed from master-slave hazard-free latches.

The area is taken as the sum of the channel areas of all the transistors.

Clock-to-Output Times

$T_{CHQV}$was estimated in two ways.

The first way was to count the number of gates that had to be traversed from D to the Q-q feedback loop. This gives a crude estimate that turned out to be surprisingly good, but don’t try this with large fanout or fanin.[1]

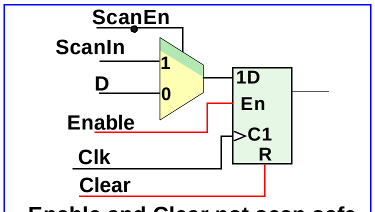
The second estimates the relative delay from the transistor capacitances as determined by their channel areas.

Neither method includes clock-to-Q delays which will shift delay from setup time to hold time.

[1] Fanout is the number of gates inputs, or in this case the capacitance, seen by an output driver. Fanin is the number of gate inputs, or more important here, the maximum number of series transistors in the gate .

▪ Scannable Flip-Flops ▪

Special Mux-Scan Flip Flop Problems



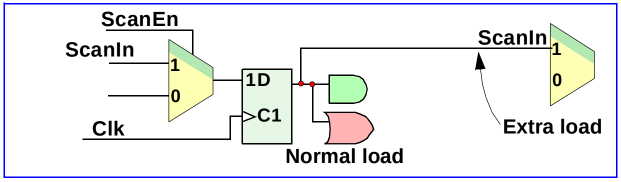
Enable and Clear not scan safe

(1) Scan Safe

Problem is FF inputs not going through scan mux

(2) Scan causes extra output loading

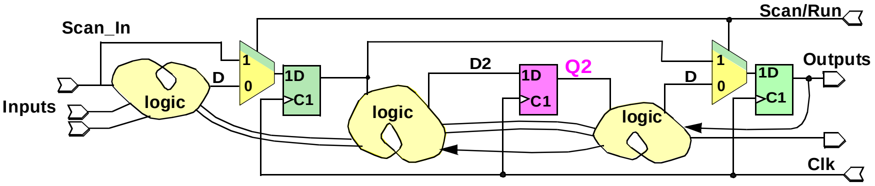
Causes a bit of extra delay



(3) For Partial Scan

Flip flops which are not scanned should hold values during scan in.

- Below, D2 has strange values during scan in.
- Thus strange values are clocked into Q2.



Special Mux Scan Flip-Flop Properties

Enable should be scan safe

Special low drive scan output.

Gated clock built in for partial scan

In the picture shown, the flip flop would be intended to run at least two clock cycles between scans.

▪ Scannable Flip-Flops ▪

Scan-Safe Mux-D Scan Flip-Flop

Scan-Safe

ENABLE, SET, or RESET (Clear) must not operate during scanning.

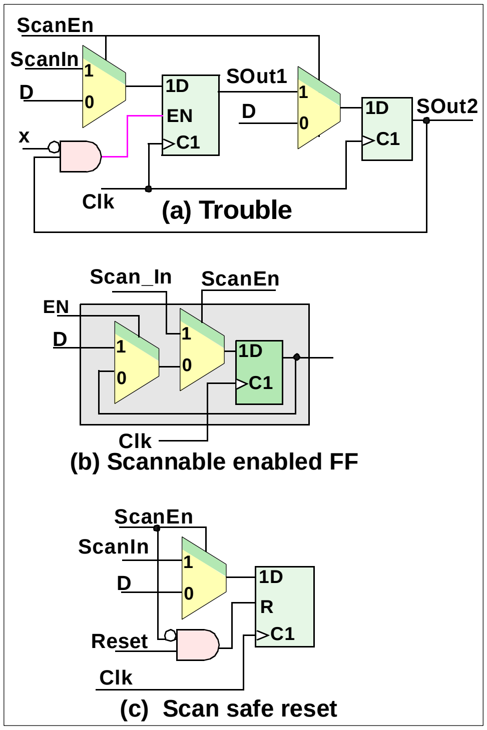
(a) Scan chain made with flip-flops and MUXs.
Enable (EN) is controlled by the scan chain.
A zero on SOut2 will shut down scanning.

(b) An enabled scannable flip-flop.
ScanEn takes precidence over EN.
Always safe.

(c) Scan safe reset.

Disconnect other signals like EN, SET and RESET when scanning.

Avoid scan pollution.



(a) Trouble

(b) Scannable enabled FF

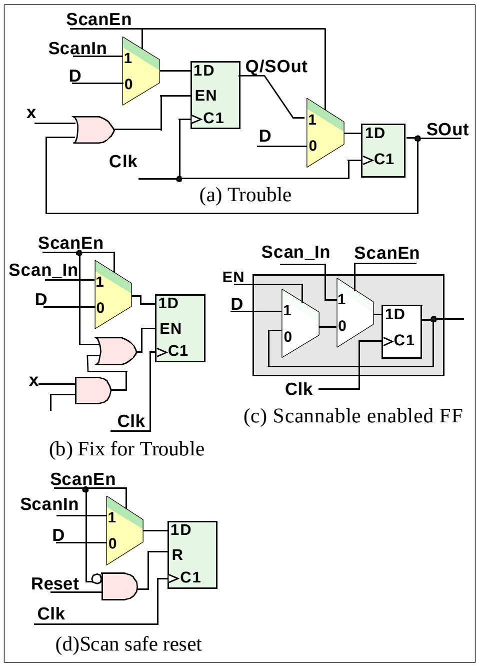
(c) Scan safe reset

The Mux-D Scan Flip-Flop

Scan-Safe

Flip-flops must be immune to all outside influences when the scan chain shift register is operating.
If the flip-flop has an enable, set, or reset which takes priority over the scan commands then one must be sure they never operate during scanning.

On the right:
(a) A scan chain made with ordinary flip-flops and MUXs. The enable (EN) is controlled by part of the scan chain and can inadvertently shut itself down.
(b) A hand coded repair that lets ScanEn control the enable.
(c) An enabled scannable flip-flop. It is scan safe. The synthesizer will use them if they are in the library.
(d) Scan safe reset.



(a) Trouble

(b) Fix for Trouble

(c) Scannable enabled FF

(d)Scan safe reset

▪ Scannable Flip-Flops ▪

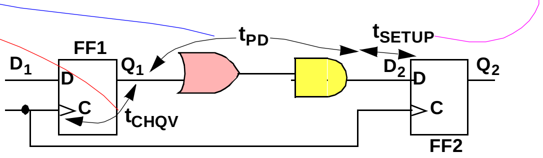
Special Scan Output

Maximum Clock Speed

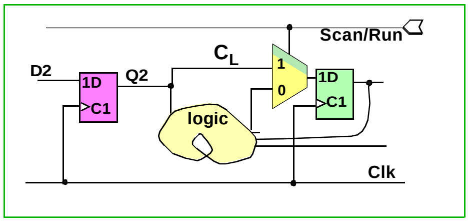
Q_1 takes t_{CHQV} to get out of FF1.
t_{PD} = propagation delay through the gate(s).
D_2 must reach FF2 at least t_{SETUP} before next clock edge.

The signal must go FF1 to FF2 in one clock cycle, thus

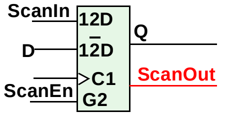
$$t_{CLOCK} \geq t_{CHQV} + t_{PD} + t_{SETUP}$$



The MUX input adds extra load C_L
This slows down t_{CHQV}
For high clock speeds, may be critical

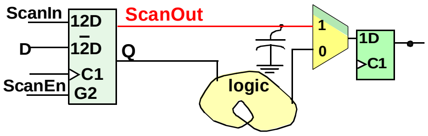


Add a parallel output for ScanOut



Weak ScanOut
separate from Q

Reduces load on Q
and speeds up t_{CHQV}

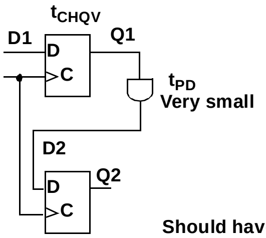


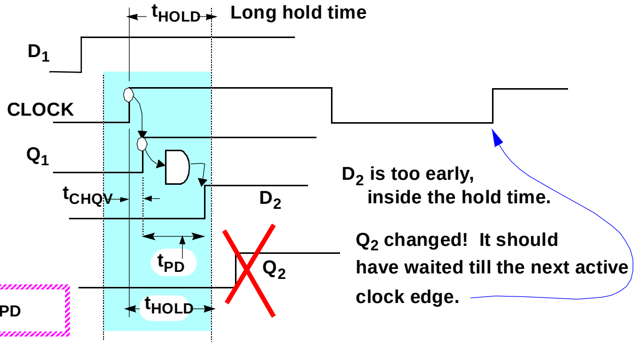
Special Scan Output

Increase t_{CHQV} for Shift Registers Scan Chains

Problem for short delays if flip-flop *hold time* > t_{CHQV}.
Then D1 can flip the upper flip-flop Q1,
dash through the gate and reach the lower flip-flop
inside its hold time!

Then both flip-flop may change on the same clock edge!





t_{HOLD}
Long hold time
D_1
CLOCK
Q_1
t_{CHQV}
D_2
t_{PD}
Q_2
t_{HOLD}
D_2 is too early, inside the hold time.
Q_2 changed! It should have waited till the next active clock edge.

Should have

$$t_{HOLD} < t_{CHQV} + t_{PD}$$

Shift Registers have very small t_{PD} .

Scan circuits are shift registers.

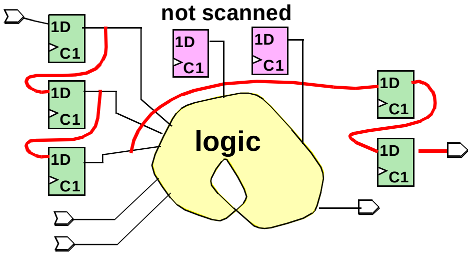
Good to have t_{CHQV} slow in shift registers.

Moral: When adding an extra scan output, make it weak and slow.

▪ Scannable Flip-Flops ▪

Gated Clocks for Partial Scan

Partial scan is when some FF are not scanned



Unscanned FF must hold old values

Mix of Mux-scan and nonscan flip-flops

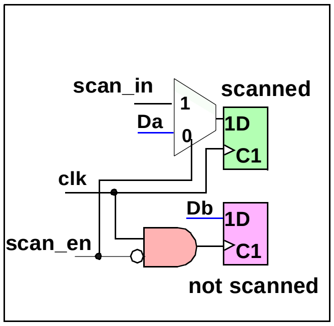
Must hold nonscan flip-flops constant during scan in.

So we will know their content during the test cycle.

Gated Clocks for Partial Scan

Synopsys gates the clock to nonscan flip-flops and latches.

Clocked scan or LSSD flip-flops
  Stop the data clock
  This avoids clock gating circuitry

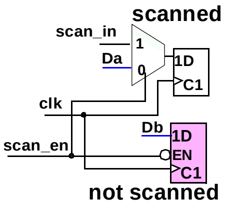


Partial Scan

Circuits with some unscanned flip-flops or latches.

Latches are used for skew correction.
Clock dividers have very critical timing and may not want to have scan added.
Full-scan may be considered too costly in area or delay,
although partial scan is often considered too complicated.

Test insertion programs

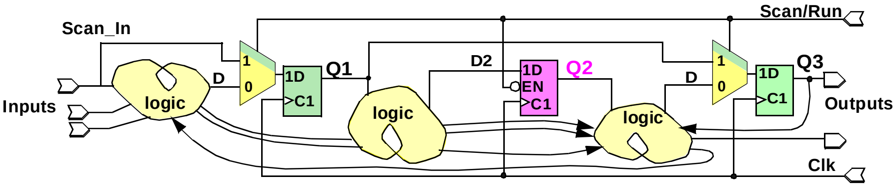


They may add add a slave latch to latches. This is unused normally but is used during scan to make the latch into a flip-flop.

For unscanned flip-flops, some test insertion programs gate the clock. This allows the flip-flop to maintain its old value during scan. A better method would be to disable the flip-flop during scan as shown.

Example of a circuit equiped for partial scan

It can: (1) scan in a test vector.
(2) Run a test which will place a desired value on D2.
(3) Scan in another test vector to put desired values on Q1 and Q3
(4) Run the this test vector which includes Q2 as an input to the combinational logic.
(5) Scan out the results.



Other Testing Problems

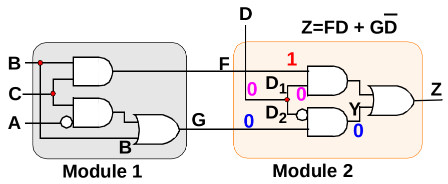
Seperately Minimized Logic Blocks

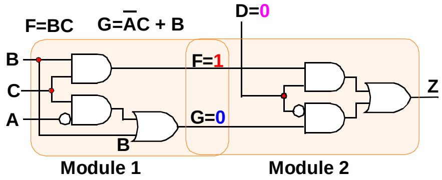
Two modules <u>synthesized seperately</u>.

(i) In module 2. F, G and D are the primary inputs;
find all the test vectors for D_1 s@1.

(ii) Take the modules together so A, B, C and D are the primary inputs.
Find a test vector for D_1 s@1 if there is one.

(iii) Why are there no test vectors?





D \ FG	00	01	11	10
0				TV
1				

D1=0 => D=0

Y=0 => G=0

F=1

TestVector D,F,G=0,1,0

AB \ CD	00	01	11	10
00				
01			1	1
11			1	1
10				

F=BC

AB \ CD	00	01	11	10
00	0	0	$\overline{A}C$	
01	B			
11				
10	0	0	0	0

G=$\overline{A}$C + B

Minimized together $Z = (BC)D +(\overline{A}C+B)\overline{D}$

$= BC + \overline{A}C\overline{D} + B\overline{D}$

Finding a Test for D_1 S@1

One Reason For Lower Test Coverage

When connected logic modules are synthesized separately, the complete circuit may not be minimized. There may be connections fed from the first to the second module which may cause problems.

The logic in module two was synthesized as though all possible combinations could appear on wires F and G. In reality only F,G = 0,0; 0.1; and 1,1 can appear. F,G=1,0 can never be generated by module one.

The test for D_1 s@1 is trying to test logic for D,F,G= 0,1,0 which can never happen in the combined module. The circuit has extra redundant logic which is never used.

If one minimized both modules together, the minimal sum-of-product's circuit would have been

$Z= BC + \overline{A}C\overline{D} + B\overline{D}$

This his no redundant logic and would have been 100% testable for single stuck-at faults.

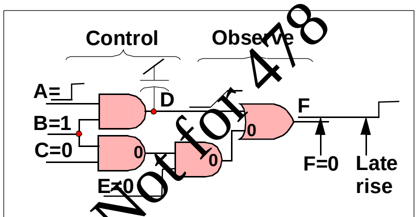
Faults for At Speed Testing

Transition Delay Faults (Gate Delay Faults)

Faults are *slow to rise* and
slow to fall

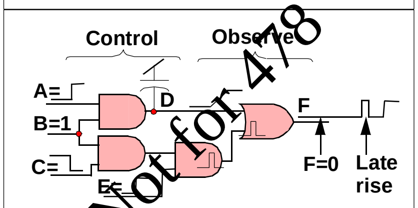
To test for *slow to rise* at D.

Force D to 0, same pattern as for a S@1 fault.
Wait until the output receives the good value.
Change pattern, force D to 1
Keep the path D to F sensitized.



Test for slow to rise at D

)Do not cause glitches in the path D to F by changing the way it is sensitized.



Improper test for slow to rise at D
Path sensitization changes

Delay fault testing always finds s@ faults.
These effectively have infinite delay.

ATPG tests are usually generated to be observed on the closest output.

Path delay tests want the longest path.
The one most likely to fail by say slightly thin metal.

In *transition delay faults* (*gate delay faults*) one wants to sensitise a path from the input to the output. ATPG for this is almost the same as for a stuck at fault test.

Transition Delay Faults (Gate Delay Faults)

▪ Other Testing Problems ▪