1. atpg

1.1 Transition AC

DC的时候scan_en为0时，只有一拍时钟，scan_en下降沿与这一拍时钟的距离可以足够远，这个时钟也可以是高速或低速时钟。

AC的时候scan_en为0时，有两拍高速时钟（这里也可以配置为低速时钟，shift_as_capture mode，

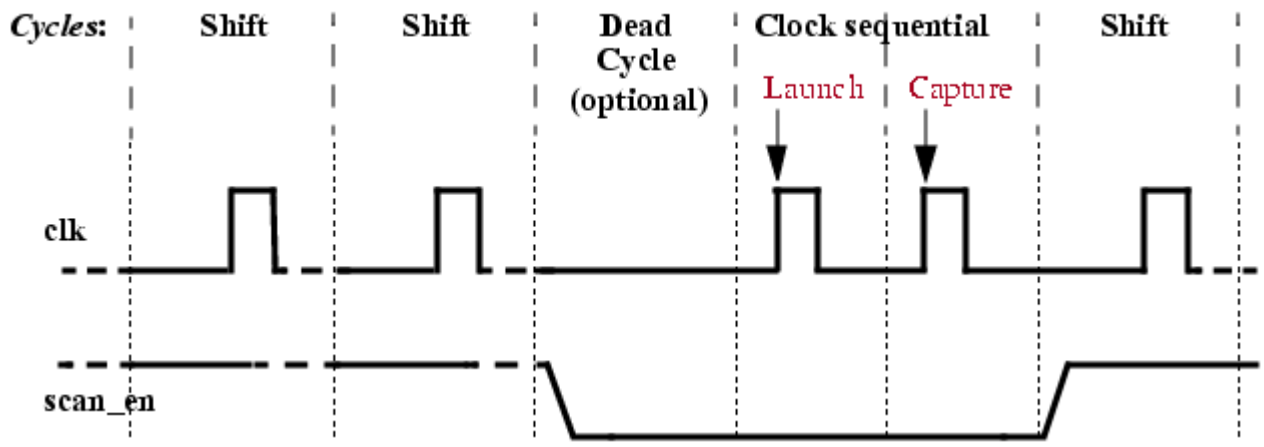
AC有两种样本方式，LOC(比较常用)和LOS，

1.1.1 LOC

https://www.cnblogs.com/6y4z/p/16588760.html

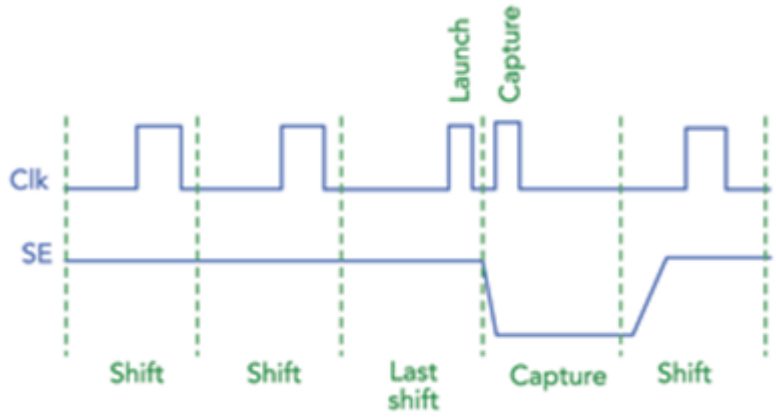
Broadside (Launch-Off-Capture)

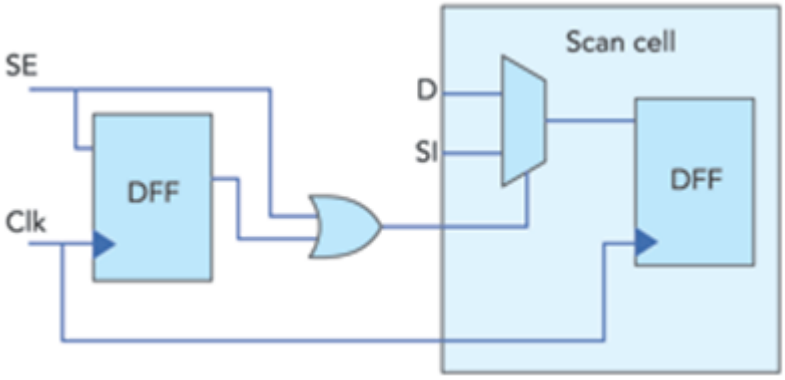
一般都是使用的这种结构，使用标准OCC即可生成期望时序。

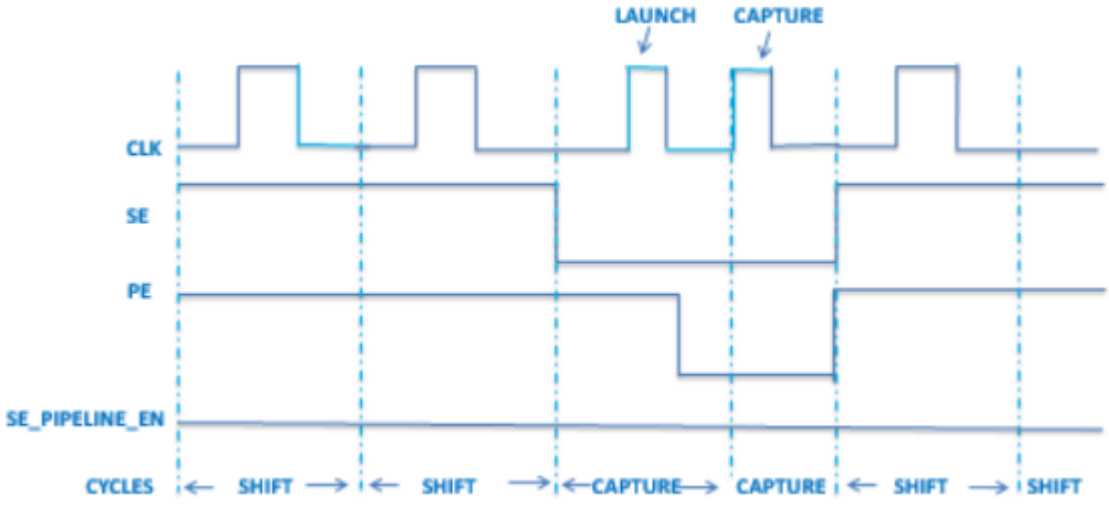


1.1.2 LOS

pseudo-Launch-Off-Shift (LOS)







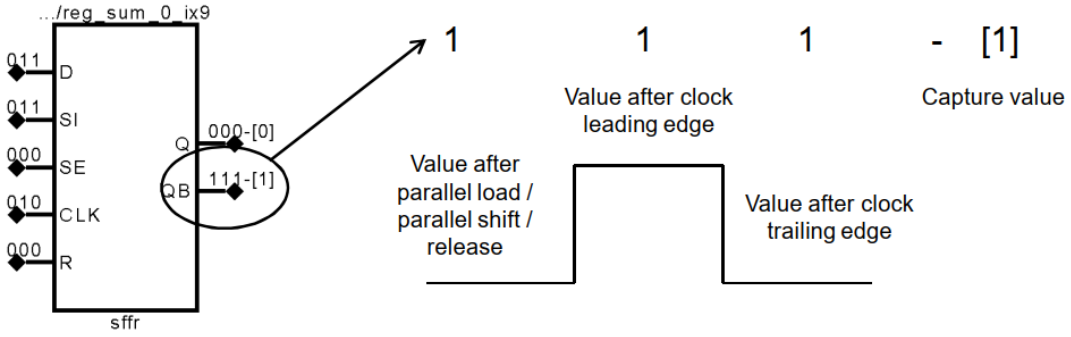
一直为1，开启se pipeline enable

se_pipeline_en一直为1，开启se pipeline enable

这里的launch & capture都是高速时钟，可以理解为跟LOC一样，是OCC生成的□OCC在这看和LOC是一样的行为，只是lauch值的区别。

1.2 pattern debug

Displayed data shows data during capture cycle clock frames of the reported pattern



1.3 Pre-Existing Scan Chains

```
>add_scan_group grp1 chain_trace.testproc
>add_scan_chains -internal my_chain0 grp1 \
  corea_gate1_tessent_edt_c1_inst/edt_scan_in[0] \
  corea_gate1_tessent_edt_c1_inst/edt_scan_out[0]
>add_scan_chains -internal my_chain1 grp1 \
  corea_gate1_tessent_edt_c1_inst/edt_scan_in[1] \
  corea_gate1_tessent_edt_c1_inst/edt_scan_out[1]
```

这种会有一个问题， 如果int_edt_mode和ext_edt_mode的chain有走部分公共路径，最后通过int/ext mode来区别chain如何分岔， 这种不知道在drc trace scan chain时怎么处理。

1.4 PR网表层次保持

PR后网表对OCC的hier层次需要固定不要动，不然在推ATPG时会找不到相应path报错。

具体发现以下在TCD文件里面描述的OCC以下PATH保持不变即可。

```
ClockOut()
ShiftRegisterClockEn()
```

1.5 OCC clone

或者是多个OCC设置兼容模式

方式一：

```
# 将同源或同频的 OCC 输出时钟分配到同一同步组
add_synchronous_clock_group {clk1 clk1_p clk2 clk2_p}
```

```
# 如果是想把OCC输出也弄成一个group的话，这里只需要写occ的inst hier name
```

方式二：
将多个clk添加同一个clock_intercept_node上，它们会共享OCC的shift cgc，永远是同步输出capture clock。

```
set occ [ add_config_element OCC/Controller($id) ]
set_config_value clock_intercept_node -in $occ $clk
```

在ATPG时启用兼容时钟模式：
set_clock_restriction domain_clock -compatible_clocks_between_loads on

2. gen atpg pattern

需要读取icl pdl, tcd, netlist文件，这样就可以跑出atpg pattern

```
//  command: read_design gps_baseband -design_id gate -verbose
//  sub-command: set_tool_options -reapply_settings_after_reelaboration On
//  sub-command: set_read_design_tag gps_baseband
//  sub-command: read_verilog
../tsdb_outdir/dft_inserted_designs/gps_baseband_gate.dft_inserted_design/gp
s_baseband.vg -no_duplicate_modules_warnings
//  sub-command: set_read_design_tag ""
//  sub-command: set_tool_options -reapply_settings_after_reelaboration Off
//  sub-command: read_icl
../tsdb_outdir/dft_inserted_designs/gps_baseband_gate.dft_inserted_design/gp
s_baseband.icl -skip_child_blocks -no_notes
//  sub-command: source
../tsdb_outdir/dft_inserted_designs/gps_baseband_gate.dft_inserted_design/gp
s_baseband.pdl
//  sub-command: read_core_descriptions
../tsdb_outdir/dft_inserted_designs/gps_baseband_gate.dft_inserted_design/gp
s_baseband.tcd
```

2.1 set_atpg_limits

设置atpg process limits, 比如可以设置只出几条pattern，这样可以快速得到一个简单的atpg pattern，可以快速用于后续flow

```
set_atpg_limits -cpu_sec 500 -test_coverage 99.5 -pattern_count 2
report_environment
```

2.2 low power shift & capture

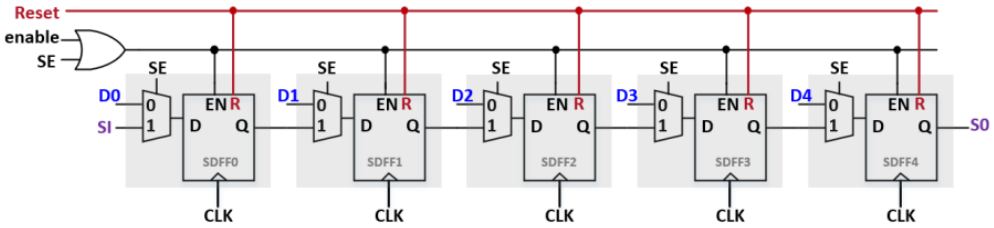


Fig. 1. Schematic design of regular scan chain with enable signal.

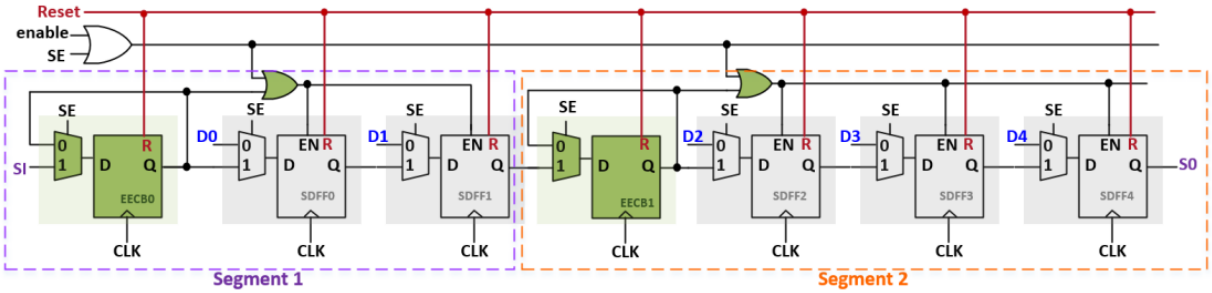


Fig. 2. Schematic design of segmented scan chain.

capture low power

用单独的几个scan cell(scan_en为0时保持cell值不变)，控制在capture阶段register的时钟enable
这样可以分段控制部分register在capture阶段数据不变，实现low power效果

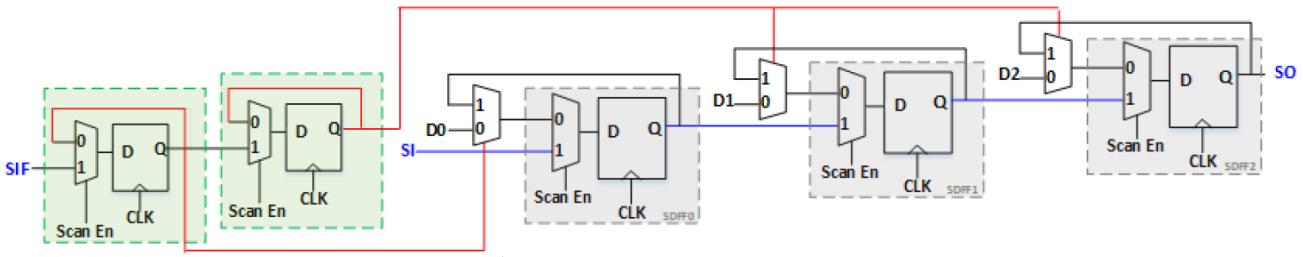


Fig. 1. Schematic design of segmented scan chain.

这个图如果控制时钟不产生的话，就和上图效果差不多了。

shift low power

主要是在shift阶段Q到function组合逻辑之间的通路加一个and/or门
让逻辑结合部分在shift阶段状态不变化,在capture阶段再变化。