Limitations:

- 32b access (easy to enhance)
- 100MHz

Don't cares:

- ODT constant
- DQS-out (FPGA to SDRAM) single ended
- DQS-in (SDRAM to FPGA) skipped

Testbench: rtl\testbench\simple stimulus

Design: rtl\system\fpga_top: Executes test sequence on FPGA
Instantiates clock generator 100MHz => 100MHz
and SDRAM_CNTRL
rtl\system\SDRAM_CNTRL:
Generates proposed refresh mechanism
(Could hold initialization phase generator)
Instantiates PHYIO
rtl\system\SDRAM_PHYIO:
PHY: Instantiates DDR IOB registers
IO Handling FSM:
Initialization phase
SDRAM access control

Constraints: constr\nexys4_ddr_sdram.xdc:
Constraints for SDRAM interface on NEXYS 4 DDR
constr\fpga_top.xdc: Project specific constraints.

Scripts: script\report_io_timing.tcl: Script to generate IO timing report

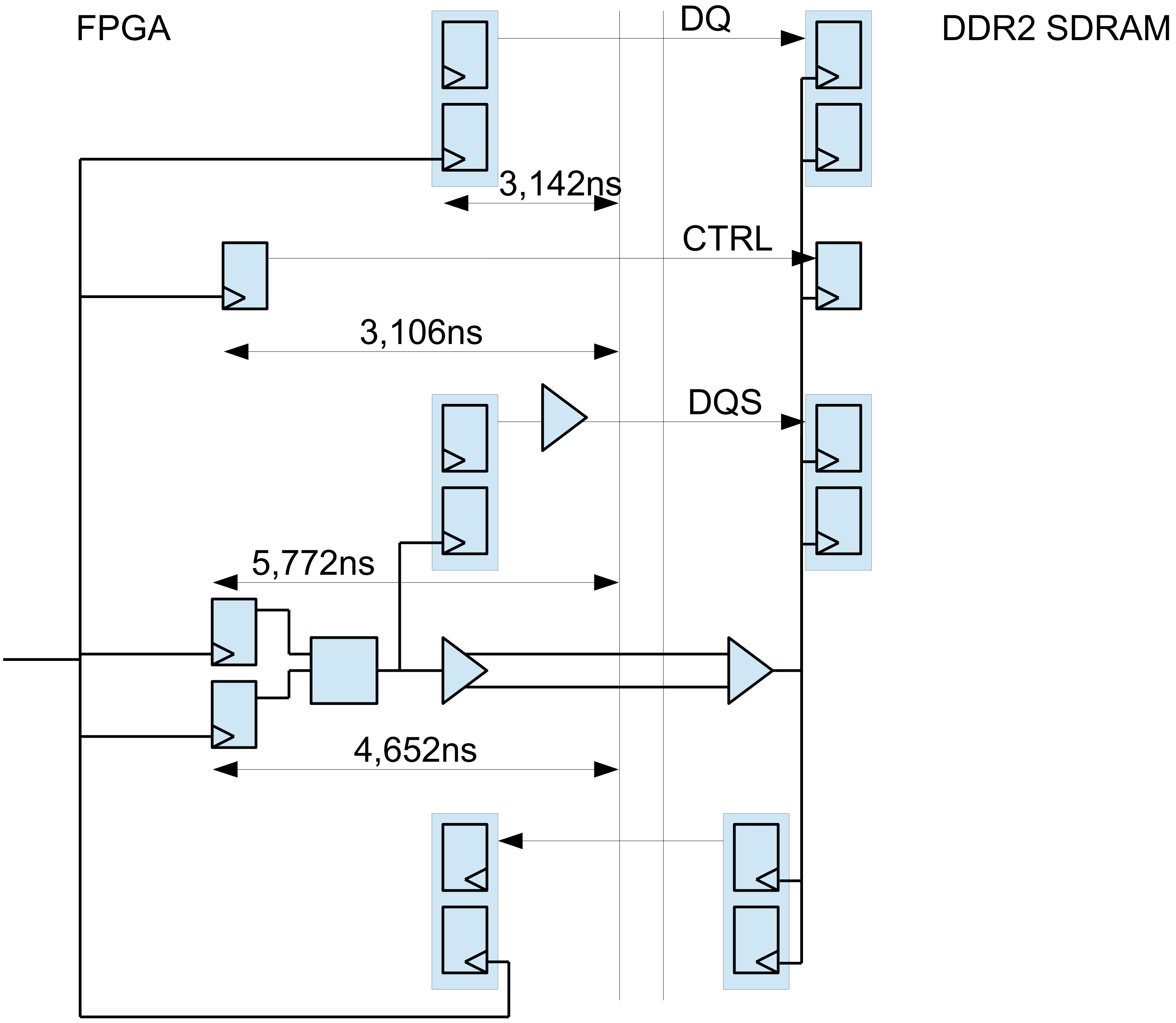
Test sequence:

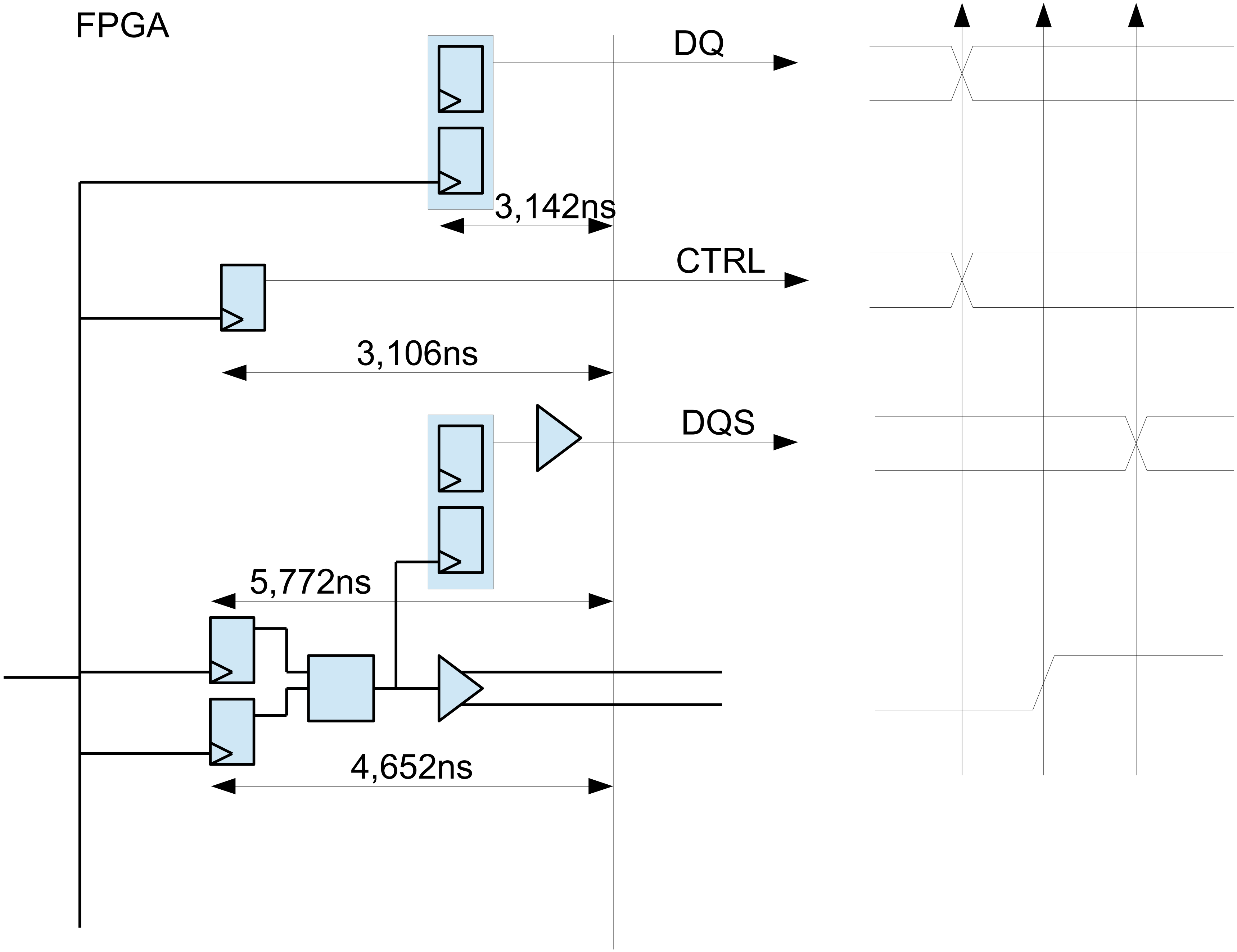
Initialization phase
Some basic writes and reads
Writes and reads to specific area (let's call it area 3 because it is on bank 3)
Loop over writes and reads throughout some banks.

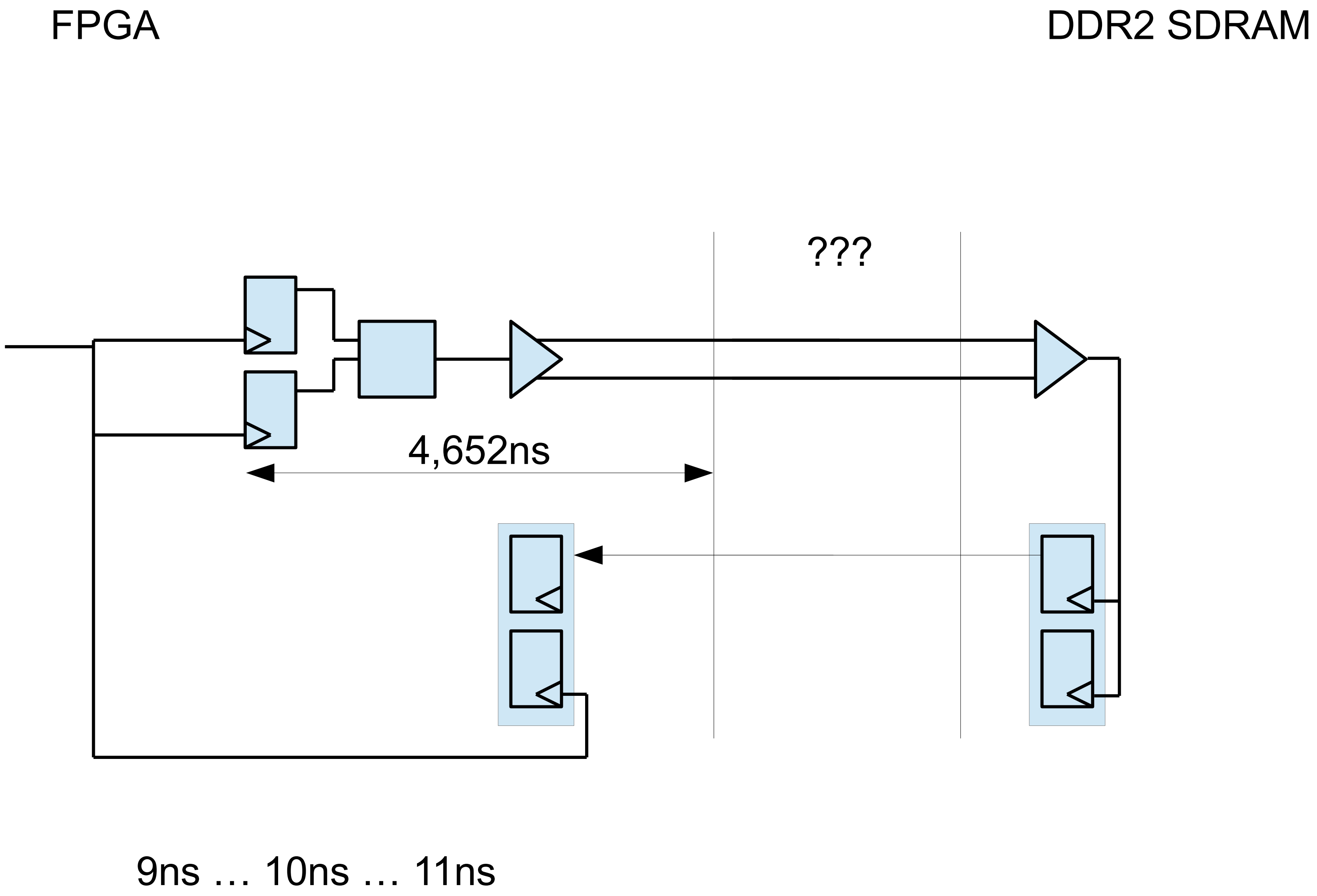
If everything works fine, led(1) turn on.

In general, if a bug was found, led(1) turns off and led(0) blinks fast (every second).

After ca. 1 minute, area 3 is read again to check, if the refresh mechanism worked. If yes, led(0) toggles to indicate this 1 minute interval and that the testcase is still active.


FPGA
DQ
DDR2 SDRAM
3,142ns
CTRL
3,106ns
DQS
5,772ns
4,652ns


FPGA
DQ
3,142ns
CTRL
3,106ns
DQS
5,772ns
4,652ns


FPGA
DDR2 SDRAM
???
4,652ns
9ns … 10ns … 11ns

Optimizations:

64b, 128b
Bytewise write
Faster clock
Faster access, multiple active banks
…
More standalone testing

=> PDVL