Total Ionizing Dose Test Report

No. 15T-RTAX250S-CQ352-D59RQ1

January 15, 2015

Table of Contents

TOTAL IONIZING DOSE TEST REPORT

No. 15T-RTAX250S-CQ352-D59RQ1

January 15, 2015

Solomon Wolday and Alison Ma

(408) 643-6388, (408) 643-6602

solomon.wolday@microsemi.com, alison.ma@microsemi.com

I.   Summary Table

Parameter	Tolerance
1. Gross Functionality	Passed 300 krad (SiO$_2$)
2. Power Supply Current (ICCA/ICCI)	Passed 300 krad (SiO$_2$)
3. Input Threshold (VTIL/VIH)	Passed 300 krad (SiO$_2$)
4. Output Drive (VOL/VOH)	Passed 300 krad (SiO$_2$)
5. Propagation Delay	Passed 300 krad (SiO$_2$) for 10% degradation criterion
6. Transition Time	Passed 300 krad (SiO$_2$)

II.   Total Ionizing Dose (TID) Testing

This testing is designed on the basis of an extensive database (see, for example, TID data of antifuse-based FPGsA at http://www.klabs.org and http://www.microsemi.com/soc) accumulated from the TID testing of many generations of antifuse-based FPGAs.

A.   Device-Under-Test (DUT) and Irradiation Parameters

Table 1 lists the DUT and irradiation parameters. During irradiation and annealing, each input or output is tied to either the ground or VCCI with a jumper.

Table 1 DUT and Irradiation Parameters

| Part Number	RTAX250S
Package	CQ352
Foundry	United Microelectronics Corp.
Technology	0.15 µm CMOS
DUT Design	TOP_AX250S_TID
Die Lot Number	D59RQ1
Quantity Tested	5
Serial Number	300 krad(SiO2): 7891, 7904, 7929 200 krad(SiO2): 7937, 7959
Radiation Facility	Defense Microelectronics Activity
Radiation Source	Co-60
Dose Rate (±5%)	7.5 krad (SiO$_2$)/min
Irradiation Temperature	Room
Irradiation and Measurement Bias (VCCI/VCCA)	Static at 3.3 V/1.5 V
I/O Configuration	Single ended: LVTTL Differential pair: LVPECL

B.   Test Method
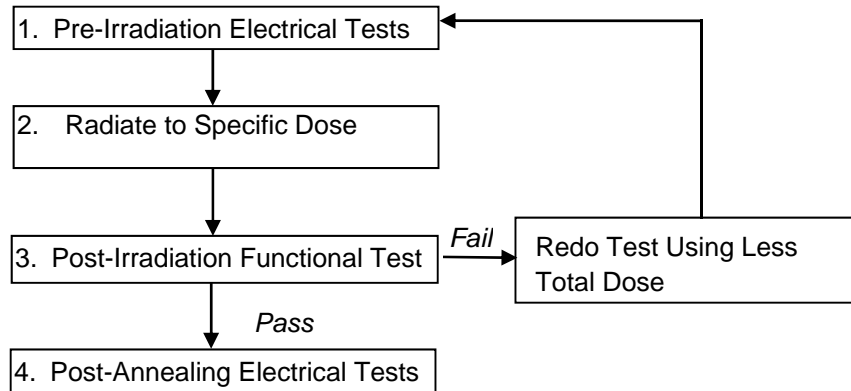
Figure 1   Parametric Test Flow Chart

The test method generally follows the guidelines in the military standard TM1019.8. Figure 1 is the flow chart showing the steps for parametric tests, irradiation, and post-irradiation annealing.

The accelerated aging, or rebound test mentioned in TM1019.8 is unnecessary because there is no adverse time-dependent effect (TDE) in Microsemi SoC Products Group products manufactured by sub-micron CMOS technology. Elevated temperature annealing actually reduces the effects originated from radiation-induced leakages. As indicated by testing data in the following sections, the predominant radiation effects in RTAX250S are due to radiation-induced leakages.

Room temperature annealing is performed in this test; the duration is approximately 7 days.

C.   Design and Parametric Measurements

Each DUT uses a high utilization generic design (RTAX250S_TID) to test total dose effects in typical space applications. Appendix A contains the schematics and Verilog files for the logic design.

Table 2 lists each electrical parameter to be measured and the corresponding logic design. The functionality is measured on the output pin (O_BS) of a combinational buffer-string with 2,300 buffers, output pins (O_ANDP_CLKF, O_ORP_CLKF, O_FF_CLKF, O_ANDC_CLKF, O_ORC_CLKF, O_ANDP_CLKG, O_ORP_CLKG, O_FF_CLKG, O_ANDC_CLKG, O_ORC_CLKG, O_ANDP_CLKH, O_ORP_CLKH, O_FF_CLKH, O_ANDC_CLKH, O_ORC_CLKH, O_ANDP_HCLKA, O_ORP_HCLKA, O_FF_HCLKA, O_ANDC_HCLKA, and O_ORC_HCLKA) of four (4) shift registers with 1,312 bits total, and half of the output pins (OUTX0, OUTX1, OUTX2, OUTX3, OUTX4, OUTX5, OUTX6 and OUTX7) of the embedded RAM configured as 2048x6.

ICC is measured on the power supply of the logic-array (ICCA) and I/O (ICCI) respectively. The input logic threshold (VIL/VIH) is measured on single-ended inputs EN8, DA, IO_I_0, IO_I_1, IO_I_2, IO_I_3, IO_I_4 and IO_I_5, and also on differential inputs Din_P_0, Din_P_1, Din_P_2, Din_P_3, Din_P_4, Din_P_5 and Din_P_6. The differential inputs are configured as LVPECL instead of LVDS. Because LVPECL uses 3.3 VDC, higher than 2.5 VDC used by LVDS, it is the worst case among the differential pairs. During the measurement on the differential inputs, the N (negative) side of the differential pair is biased at 1.8 V. The output-drive voltage (VOL/VOH) is measured on QA_0. The propagation delay is measured on the output (O_BS) of the buffer string. The definition is the time delay from the triggering edge at the CLOCK input to the switching edge at the output O_BS. Both the delays of low-to-high and high-to-low output transitions are measured; the reported delay is the average of these two measurements. The transition characteristics, measured on the output O_BS, are shown as oscilloscope snapshots.

Table 2 Logic Design for Parametric Measurements

Parameters	Logic Design
1. Functionality	All key logic functions (O_BS, O_ANDP_CLKF, O_ORP_CLKF, O_FF_CLKF, O_ANDC_CLKF, O_ORC_CLKF, O_ANDP_CLKG, O_ORP_CLKG, O_FF_CLKG, O_ANDC_CLKG, O_ORC_CLKG, O_ANDP_CLKH, O_ORP_CLKH, O_FF_CLKH, O_ANDC_CLKH, O_ORC_CLKH, O_ANDP_HCLKA, O_ORP_HCLKA, O_FF_HCLKA, O_ANDC_HCLKA, and O_ORC_HCLKA), and outputs of embedded RAM (OUTX0, OUTX1, OUTX2, OUTX3, OUTX4, OUTX5, OUTX6 and OUTX7)
2. ICC (ICCA/ICCI)	DUT power supply
3. Input Threshold (VIL/VIH)	Single ended inputs (EN8/YQ0, DA/QA0, IO_I_0/IO_O_0, IO_I_1/IO_O_1, IO_I_2/IO_O_2, IO_I_3/IO_O_3, IO_I_4/IO_O_4, IO_I_5/IO_O_5), and differential inputs (Din_P_0/Dout_0, Din_P_1/Dout_1, Din_P_2/Dout_2, Din_P_3/Dout_3, Din_P_4/Dout_4, Din_P_5/Dout_5, Din_P_6/Dout_6)
4. Output Drive (VOL/VOH)	Output buffer (DA/QA_0)
5. Propagation Delay	String of buffers (CLOCK to O_BS)
6. Transition Characteristic	D flip-flop output (O_BS)

III. Test Results

A.   Functionality

Every DUT passed the pre-irradiation and post-annealing functional tests. The as-irradiated DUT is functionally tested on the output (O_FF_HCLKA) of the largest shift register.

B.   Power Supply Current (ICCA and ICCI)

Table 3 summarizes the pre-irradiation, post-irradiation and post-annealing ICC. The post-annealing ICC for four different bit patterns, all '0', all '1', checkerboard and inverted-checkerboard, in the RAM are basically the same value.

Table 3 Pre-irradiation, Post Irradiation and Post-Annealing ICC

DUT	Total Dose	ICCA (mA)			ICCI (mA)		
		Pre-Irrad.	Post-Irrad.	Post-Ann.	Pre-Irrad.	Post-Irrad.	Post-Ann.
7891	300 krad	4.4	24.3	5.6	17.7	65.3	28.9
7904	300 krad	6.0	43.6	6.6	17.6	110.0	29.5
7929	300 krad	4.5	44.9	6.7	17.7	77.2	28.8
7937	200 krad	4.3	55.0	4.6	17.7	80.7	25.9
7959	200 krad	5.1	12.7	5.1	17.7	177.7	26.0

In compliance with TM1019.8 subsection 3.11.2.c, the post-irradiation-parametric limit (PIPL) for the post-annealing ICCI in this test is defined as the addition of highest ICCI, ICCDA, and ICCDIFFA values in Table 2-4 of the RTAX-S/SL/DSP datasheet:

  http://www.microsemi.com/soc/documents/RTAXS_DS.pdf

For ICCA, the PIPL is 250 mA; the PIPL of ICCI equals $20 + 10 + 3.13 \times 7 = 51.91$ (mA). Note that there are 7 pairs of differential LVPECL inputs in each DUT. Based on these PIPL, each post-annealed DUT passes both the ICCA and ICCI spec for 300 krad(SiO_2).

C. Single-Ended Input Logic Threshold (VIL/VIH)

Table 4a through Table 4d list the pre-irradiation and post-annealing single-ended input logic threshold.
All data are within the specification limits, and the post-annealing shift in every case is less than 10%.

Table 4a Pre-Irradiation and Post-Annealing Input Thresholds

In/Out Pin:		IO_I_0/IO_O_0				IO_I_1/IO_O_1			
DUT	Total Dose	Pre-Irrad.	Post-Ann.	Pre-Irrad.	Post-Ann.	Pre-Irrad.	Post-Ann.	Pre-Irrad.	Post-Ann.
		VIL (V)		VIH (V)		VIL (V)		VIH (V)	
7891	300 krad	1.490	1.490	1.500	1.495	1.500	1.505	1.515	1.505
7904	300 krad	1.495	1.485	1.510	1.490	1.500	1.510	1.515	1.500
7929	300 krad	1.495	1.475	1.505	1.475	1.500	1.500	1.515	1.500
7937	200 krad	1.500	1.480	1.505	1.485	1.525	1.485	1.525	1.435
7959	200 krad	1.515	1.475	1.500	1.490	1.520	1.505	1.515	1.500

Table 4b Pre-Irradiation and Post-Annealing Input Thresholds

In/Out Pin:		IO_I_2/IO_O_2				IO_I_3/IO_O_3			
DUT	Total Dose	Pre-Irrad.	Post-Ann.	Pre-Irrad.	Post-Ann.	Pre-Irrad.	Post-Ann.	Pre-Irrad.	Post-Ann.
		VIL (V)		VIH (V)		VIL (V)		VIH (V)	
7891	300 krad	1.500	1.440	1.480	1.465	1.460	1.455	1.480	1.500
7904	300 krad	1.475	1.445	1.450	1.435	1.470	1.500	1.475	1.500
7929	300 krad	1.490	1.435	1.485	1.430	1.450	1.435	1.510	1.495
7937	200 krad	1.460	1.490	1.485	1.435	1.465	1.450	1.475	1.495
7959	200 krad	1.485	1.430	1.485	1.430	1.470	1.440	1.475	1.495

Table 4c Pre-Irradiation and Post-Annealing Input Thresholds

In/Out Pin:		IO_I_4/IO_O_4				IO_I_5/IO_O_5			
DUT	Total Dose	Pre-Irrad.	Post-Ann.	Pre-Irrad.	Post-Ann.	Pre-Irrad.	Post-Ann.	Pre-Irrad.	Post-Ann.
		VIL (V)		VIH (V)		VIL (V)		VIH (V)	
7891	300 krad	1.505	1.490	1.495	1.465	1.465	1.450	1.485	1.500
7904	300 krad	1.515	1.500	1.505	1.505	1.470	1.470	1.485	1.495
7929	300 krad	1.500	1.495	1.500	1.460	1.460	1.440	1.515	1.460
7937	200 krad	1.510	1.490	1.490	1.475	1.460	1.450	1.485	1.460
7959	200 krad	1.505	1.485	1.480	1.465	1.520	1.460	1.495	1.500

Table 4d Pre-Irradiation and Post-Annealing Input Thresholds

In/Out Pin:		DA/QA0				EN8/YQ0			
DUT	Total Dose	Pre-Irrad.	Post-Ann.	Pre-Irrad.	Post-Ann.	Pre-Irrad.	Post-Ann.	Pre-Irrad.	Post-Ann.
		VIL (V)		VIH (V)		VIL (V)		VIH (V)	
7891	300 krad	1.510	1.505	1.525	1.515	1.500	1.490	1.525	1.520
7904	300 krad	1.510	1.505	1.525	1.515	1.500	1.490	1.525	1.520
7929	300 krad	1.500	1.495	1.515	1.505	1.490	1.480	1.520	1.510
7937	200 krad	1.510	1.505	1.520	1.515	1.495	1.490	1.525	1.520
7959	200 krad	1.510	1.500	1.520	1.510	1.500	1.485	1.530	1.515

Total Ionizing Dose Test Report No. 15T-RTAX250SL-LG624-D59RQ1

D.   Differential Input (LVPECL) Threshold Voltage (VIL/VIH)

Table 5a through Table 5d shows the LVPECL differential input threshold voltage changes due to irradiations. All pins show negligible changes, and all the data are well within the specification.

Table 5a Pre-Irradiation and Post-Annealing Differential Input Thresholds

In/Out Pin:		Din_P_0/Dout_0				Din_P_1/Dout_1			
DUT	Total Dose	Pre-Irrad.	Post-Ann.	Pre-Irrad.	Post-Ann.	Pre-Irrad.	Post-Ann.	Pre-Irrad.	Post-Ann.
		VIL (V)		VIH (V)		VIL (V)		VIH (V)	
7891	300 krad	1.770	1.770	1.805	1.805	1.790	1.785	1.785	1.785
7904	300 krad	1.775	1.775	1.810	1.810	1.790	1.790	1.780	1.780
7929	300 krad	1.770	1.770	1.805	1.805	1.790	1.790	1.780	1.780
7937	200 krad	1.770	1.775	1.805	1.805	1.785	1.790	1.780	1.785
7959	200 krad	1.775	1.780	1.810	1.810	1.790	1.785	1.785	1.780

Table 5b Pre-Irradiation and Post-Annealing Differential Input Thresholds

In/Out Pin:		Din_P_2/Dout_2				Din_P_3/Dout_3			
DUT	Total Dose	Pre-Irrad.	Post-Ann.	Pre-Irrad.	Post-Ann.	Pre-Irrad.	Post-Ann.	Pre-Irrad.	Post-Ann.
		VIL (V)		VIH (V)		VIL (V)		VIH (V)	
7891	300 krad	1.790	1.790	1.785	1.785	1.780	1.785	1.790	1.785
7904	300 krad	1.785	1.785	1.785	1.785	1.795	1.795	1.805	1.805
7929	300 krad	1.785	1.785	1.780	1.780	1.780	1.785	1.790	1.785
7937	200 krad	1.785	1.785	1.780	1.780	1.785	1.790	1.790	1.785
7959	200 krad	1.790	1.780	1.785	1.780	1.790	1.810	1.785	1.800

Table 5c Pre-Irradiation and Post-Annealing Differential Input Thresholds

In/Out Pin:		Din_P_4/Dout_4				Din_P_5/Dout_5			
DUT	Total Dose	Pre-Irrad.	Post-Ann.	Pre-Irrad.	Post-Ann.	Pre-Irrad.	Post-Ann.	Pre-Irrad.	Post-Ann.
		VIL (V)		VIH (V)		VIL (V)		VIH (V)	
7891	300 krad	1.790	1.790	1.800	1.800	1.715	1.785	1.805	1.780
7904	300 krad	1.790	1.790	1.800	1.800	1.715	1.810	1.800	1.760
7929	300 krad	1.785	1.785	1.795	1.795	1.780	1.735	1.790	1.800
7937	200 krad	1.785	1.785	1.795	1.795	1.790	1.775	1.780	1.785
7959	200 krad	1.785	1.790	1.795	1.800	1.810	1.745	1.780	1.855

Table 5d Pre-Irradiation and Post-Annealing Differential Input Thresholds

In/Out Pin:		Din_P_6/Dout_6			
DUT	Total Dose	Pre-Irrad.	Post-Ann.	Pre-Irrad.	Post-Ann.
		VIL (V)		VIH (V)	
7891	300 krad	1.785	1.785	1.785	1.785
7904	300 krad	1.780	1.780	1.785	1.780
7929	300 krad	1.780	1.780	1.785	1.785
7937	200 krad	1.790	1.790	1.780	1.770
7959	200 krad	1.775	1.805	1.775	1.775

E.   Output-Drive Voltage (VOL/VOH)

The pre-irradiation and post-annealing VOL/VOH are listed in Tables 6 and 7. The post-annealing data are within the specification limits; in each case, the radiation-induced degradation is well within 10%.

Table 6 Pre-Irradiation and Post-Annealing VOL (V) at Various Sinking Current

DUT	Total Dose	1 mA		12 mA		20 mA		50 mA		100 mA	
		Pre-rad	Pos-an	Pre-rad	Pos-an	Pre-rad	Pos-an	Pre-rad	Pos-an	Pre-rad	Pos-an
7891	300 krad	0.010	0.010	0.111	0.108	0.184	0.180	0.465	0.451	0.961	0.931
7904	300 krad	0.010	0.010	0.111	0.108	0.185	0.179	0.465	0.451	0.961	0.931
7929	300 krad	0.010	0.010	0.112	0.109	0.186	0.182	0.469	0.457	0.969	0.941
7937	200 krad	0.010	0.010	0.111	0.108	0.184	0.180	0.465	0.454	0.961	0.937
7959	200 krad	0.010	0.010	0.113	0.110	0.187	0.183	0.472	0.461	0.976	0.952

Table 7 Pre-Irradiation and Post-Annealing VOH (V) at Various Sourcing Current

DUT	Total Dose	1 mA		8 mA		20 mA		50 mA		100 mA	
		Pre-rad	Pos-an	Pre-rad	Pos-an	Pre-rad	Pos-an	Pre-rad	Pos-an	Pre-rad	Pos-an
7891	300 krad	3.287	3.283	3.208	3.204	3.071	3.066	2.713	2.708	2.037	2.029
7904	300 krad	3.287	3.283	3.208	3.204	3.071	3.066	2.713	2.706	2.035	2.023
7929	300 krad	3.288	3.284	3.208	3.203	3.068	3.064	2.707	2.701	2.023	2.011
7937	200 krad	3.287	3.285	3.208	3.206	3.070	3.067	2.711	2.707	2.031	2.023
7959	200 krad	3.287	3.285	3.207	3.204	3.067	3.063	2.702	2.698	2.011	2.003

F.   Propagation Delay

Table 8 lists the pre-irradiation and post-annealing propagation delays. The results show small radiation effects; in any case the percentage change is well below 10%.

Table 8 Radiation-Induced Propagation Delay Degradations

Delay (ns)

DUT	Total Dose	Pre-rad.	Post-rad.	Post-ann.
7891	300 krad	576.5	576.85	569.48
7904	300 krad	574.5	578.05	569.32
7929	300 krad	573	576.16	569.31
7937	200 krad	580.5	577.2	578.85
7959	200 krad	593.5	584.95	580.5

Radiation Δ (%)

DUT	Total Dose	Pre-rad.	Post-rad.	Post-ann.
7891	300 krad	–	0.07 %	-1.21 %
7904	300 krad	–	0.62 %	-0.9 %
7929	300 krad	–	0.56 %	-0.64 %
7937	200 krad	–	-0.56 %	-0.28 %
7959	200 krad	–	-1.44 %	-2.19 %

G.   Transition Time

Figure 2a to Figure 11b show the pre-irradiation and post-annealing transition edges. In each case, the radiation-induced transition-time degradation is not observable.
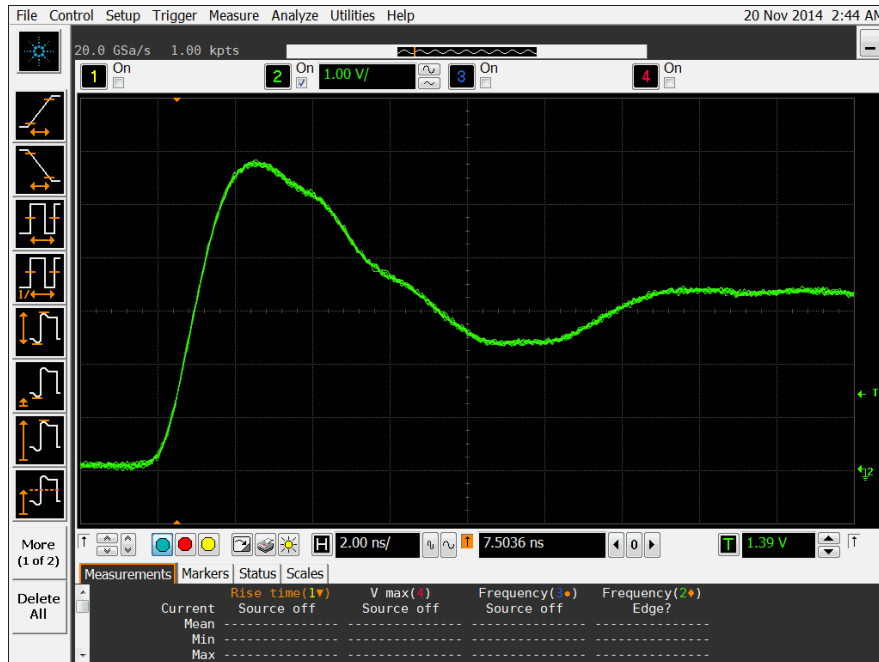
**Figure 2a DUT 7891 Pre-Irradiation Rising Edge,
abscissa scale is 1 V/div and ordinate scale is 2 ns/div.**
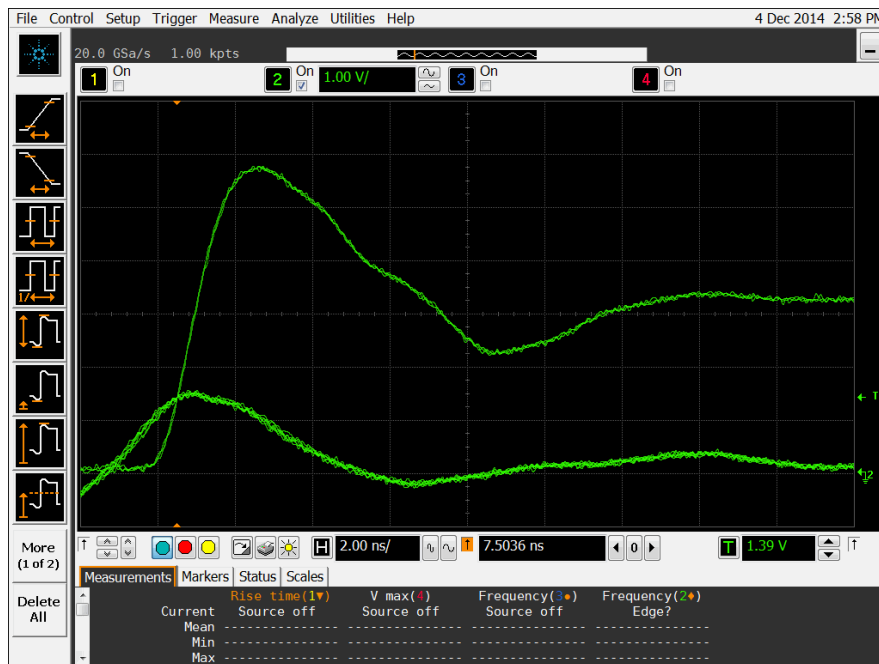
**Figure 2b DUT 7891 Post-Annealing Rising Edge,
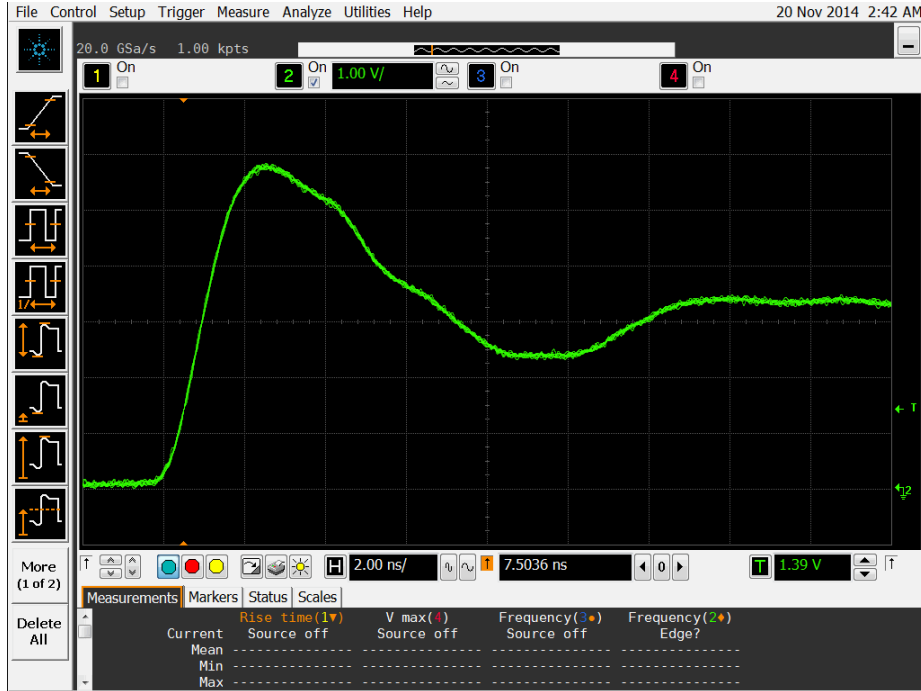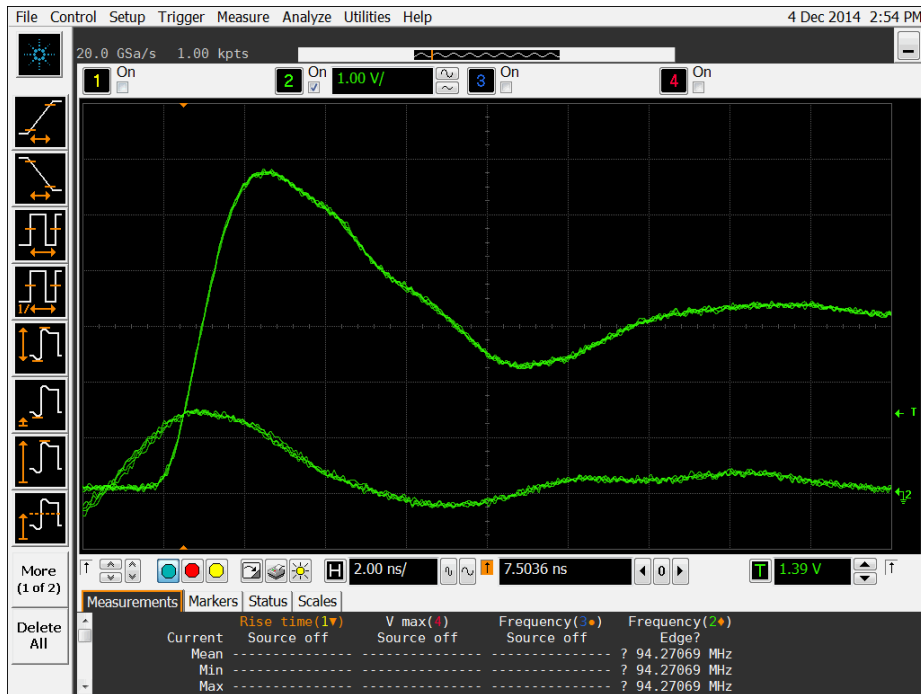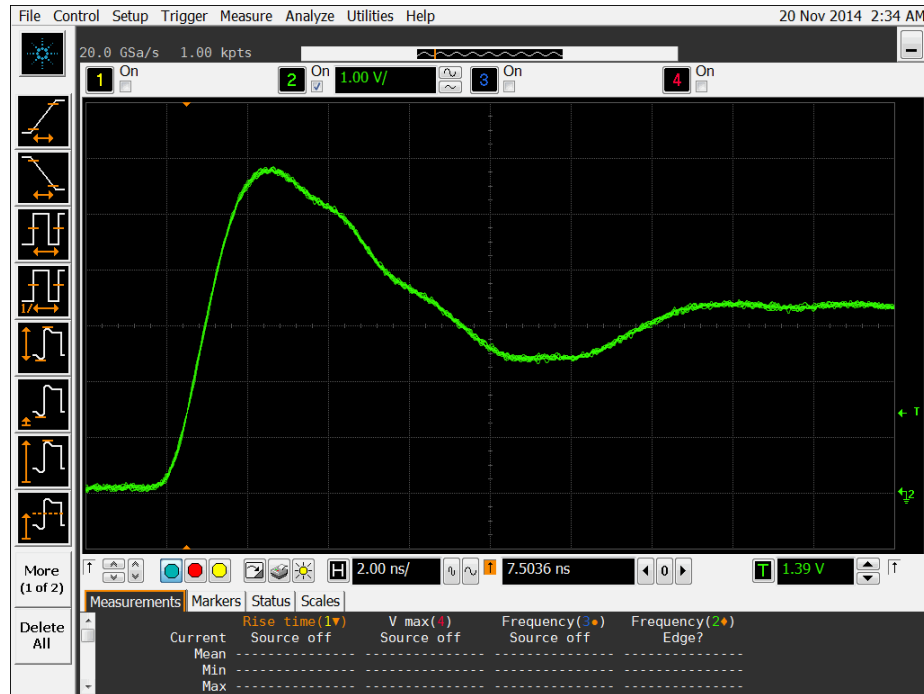abscissa scale is 1 V/div and ordinate scale is 2 ns/div.**
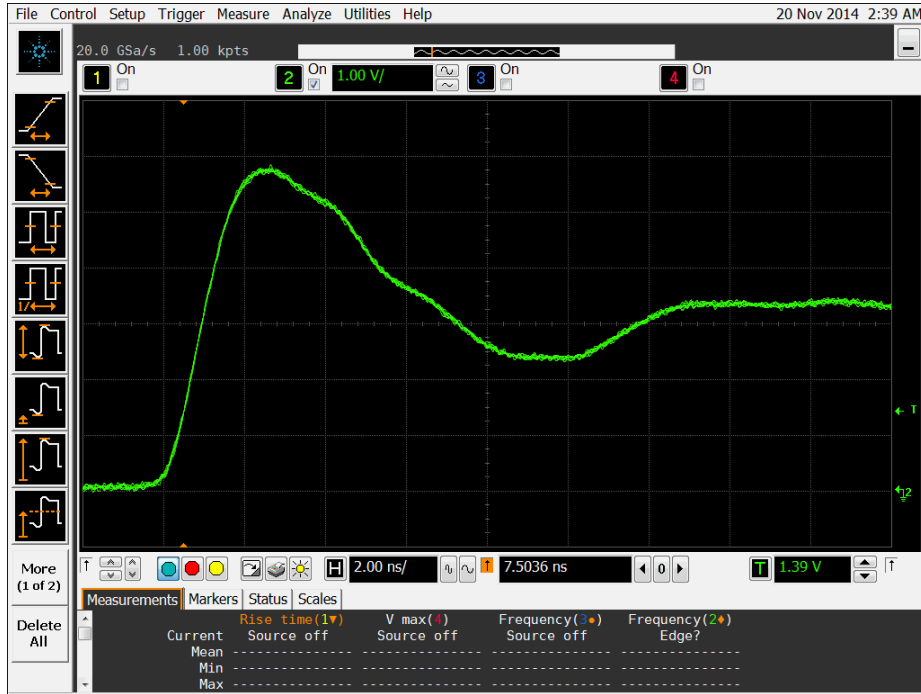
Figure 3a DUT 7904 Pre-irradiation Rising Edge,
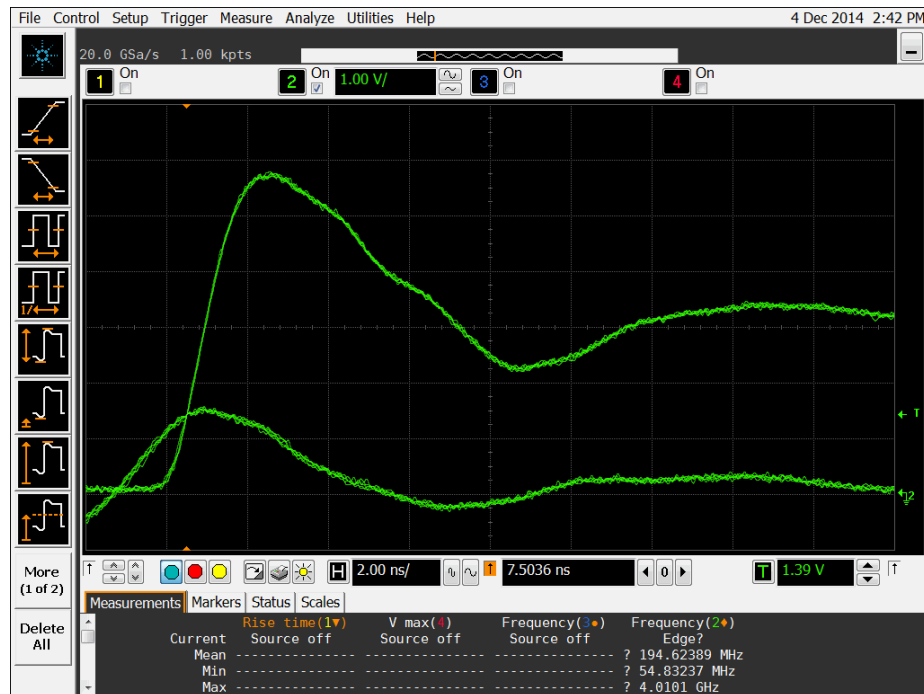**abscissa scale is 1 V/div and ordinate scale is 2 ns/div.**
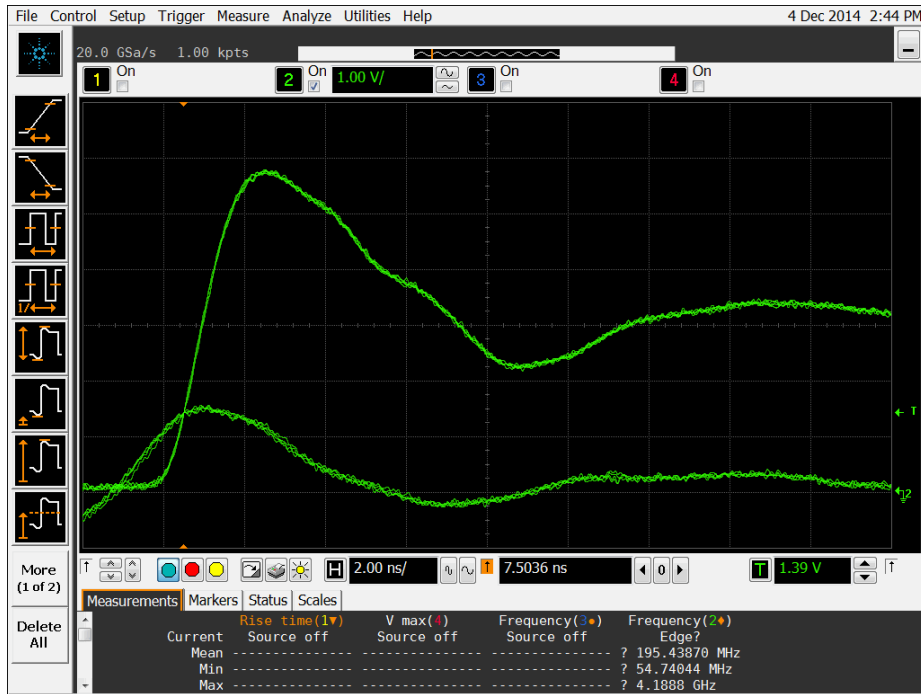
Figure 3b DUT 7904 Post-Annealing Rising Edge,
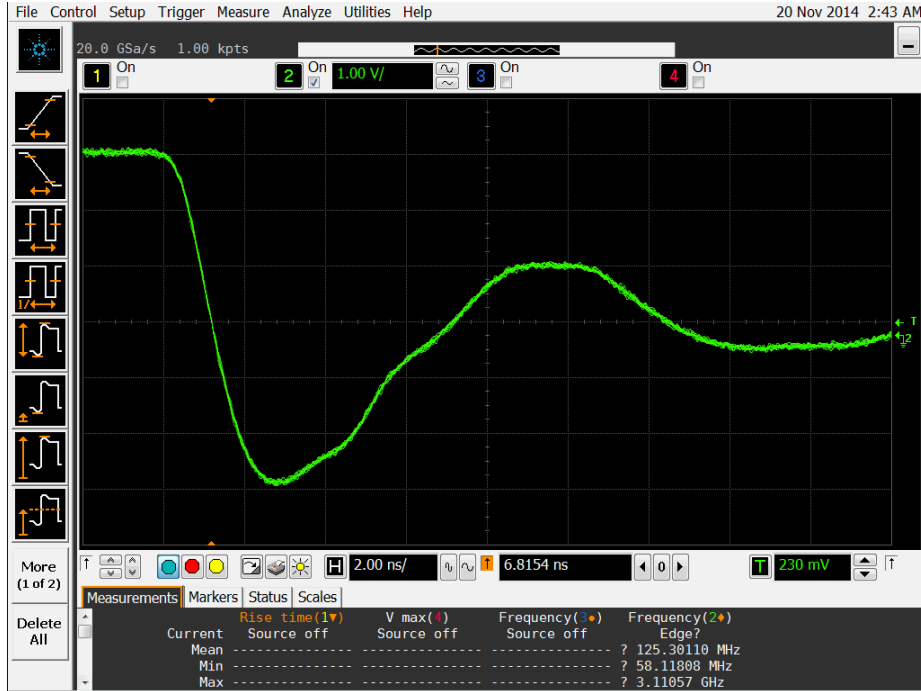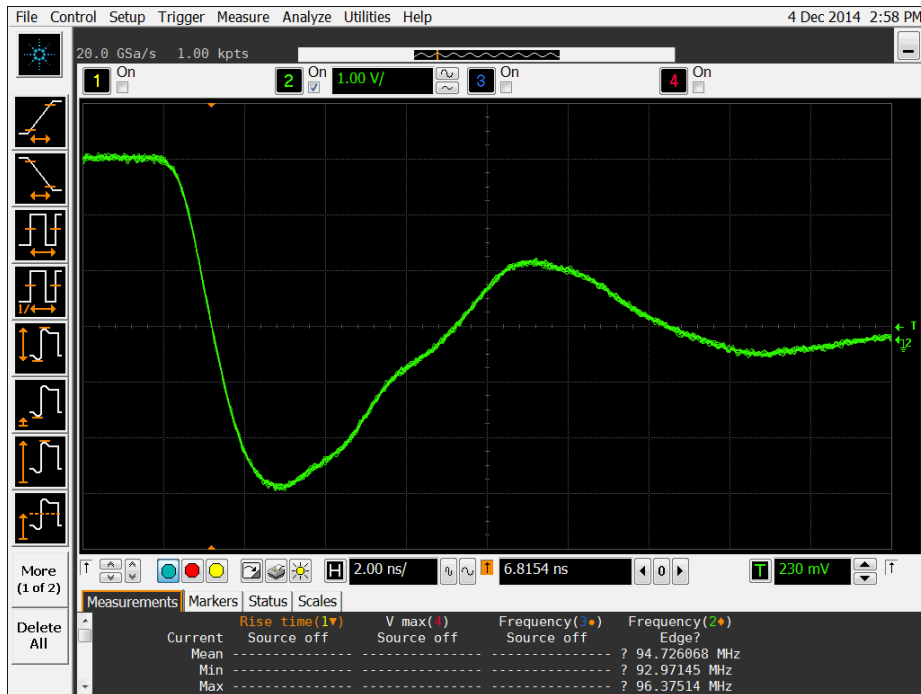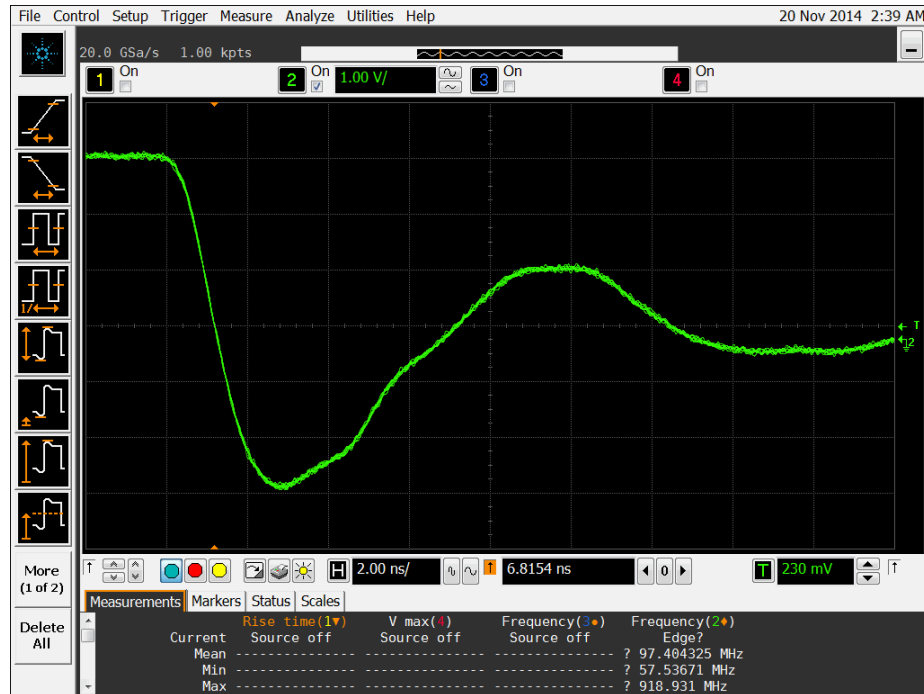**abscissa scale is 1 V/div and ordinate scale is 2 ns/div.**

File Control Setup Trigger Measure Analyze Utilities Help   20 Nov 2014 2:46 AM

20.0 GSa/s   1.00 kpts

1 On   2 On 1.00 V/   3 On   4 On

2.00 ns/   7.5036 ns   0   1.39 V

Measurements | Markers | Status | Scales

	Rise time(1▼)	V max(4)	Frequency(3●)	Frequency(2♦)
Current	Source off	Source off	Source off	Edge?
Mean	---------------	---------------	---------------	---------------
Min	---------------	---------------	---------------	---------------
Max	---------------	---------------	---------------	---------------
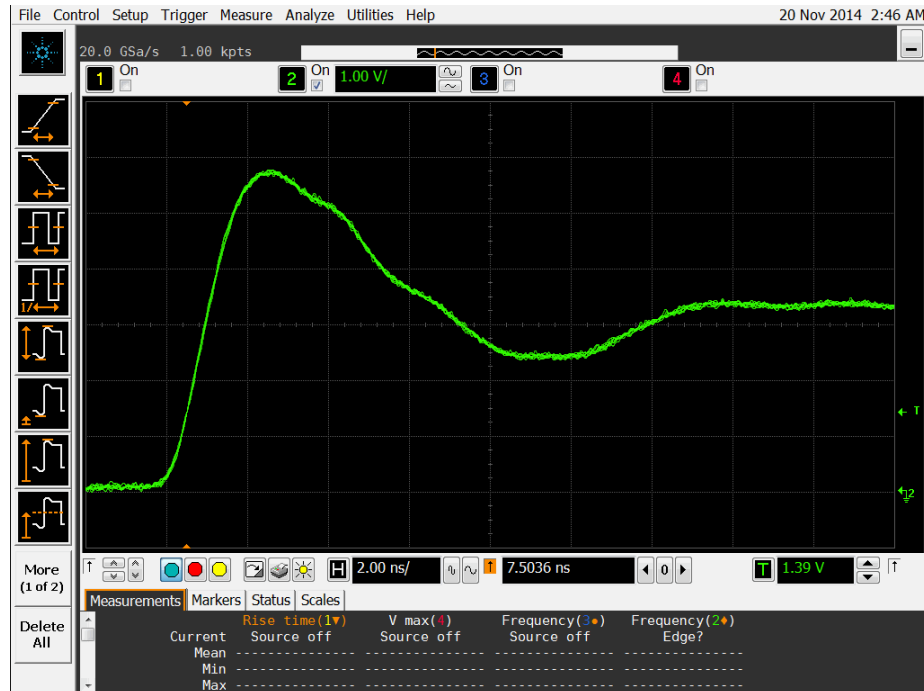
Figure 4a DUT 7929 Pre-Irradiation Rising Edge,
abscissa scale is 1 V/div and ordinate scale is 2 ns/div.

File Control Setup Trigger Measure Analyze Utilities Help   4 Dec 2014 2:59 PM

20.0 GSa/s   1.00 kpts

1 On   2 On 1.00 V/   3 On   4 On

2.00 ns/   7.5036 ns   0   1.39 V

Measurements | Markers | Status | Scales

	Rise time(1▼)	V max(4)	Frequency(3●)	Frequency(2♦)
Current	Source off	Source off	Source off	Edge?
Mean	---------------	---------------	---------------	? 203.77435 MHz
Min	---------------	---------------	---------------	? 53.01819 MHz
Max	---------------	---------------	---------------	? 6.5629 GHz
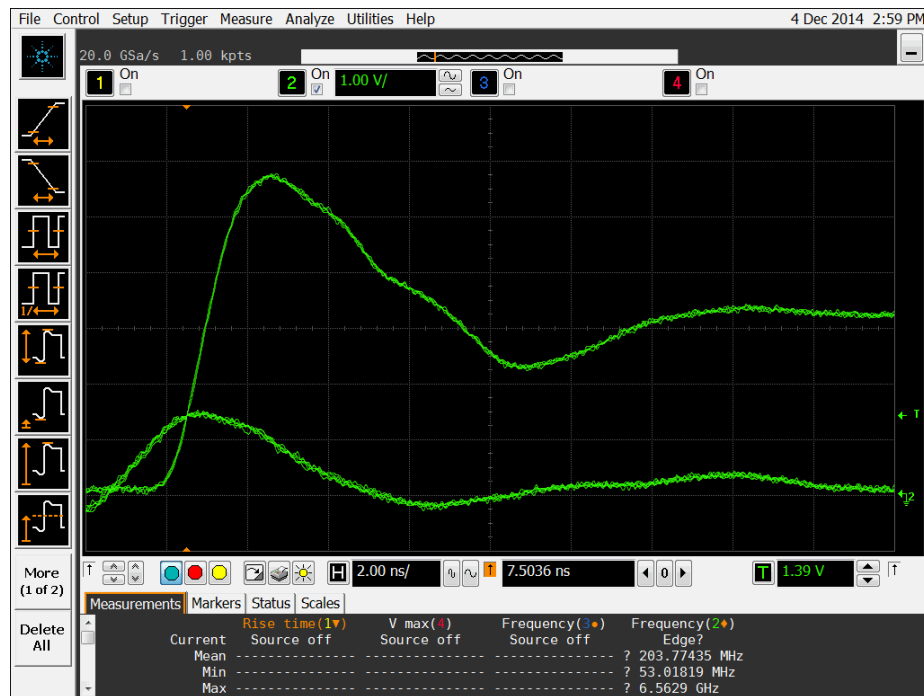
Figure 4b DUT 7929 Post-Annealing Rising Edge,
abscissa scale is 1 V/div and ordinate scale is 2 ns/div.

**Figure 5a DUT 7937 Pre-Irradiation Rising Edge,
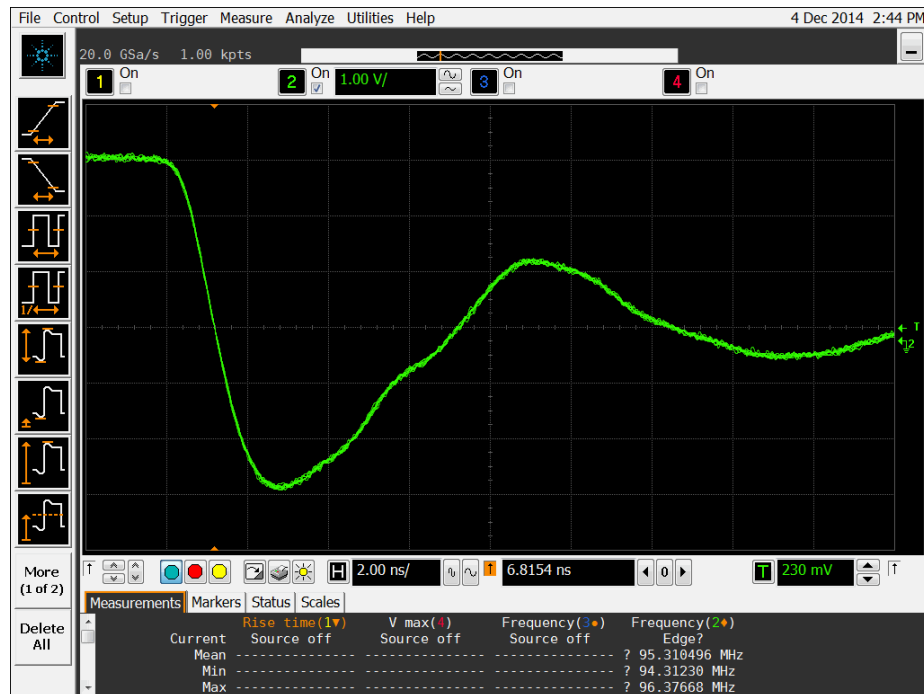abscissa scale is 1 V/div and ordinate scale is 2 ns/div.**

**Figure 5b DUT 7937 Post-Annealing Rising Edge,
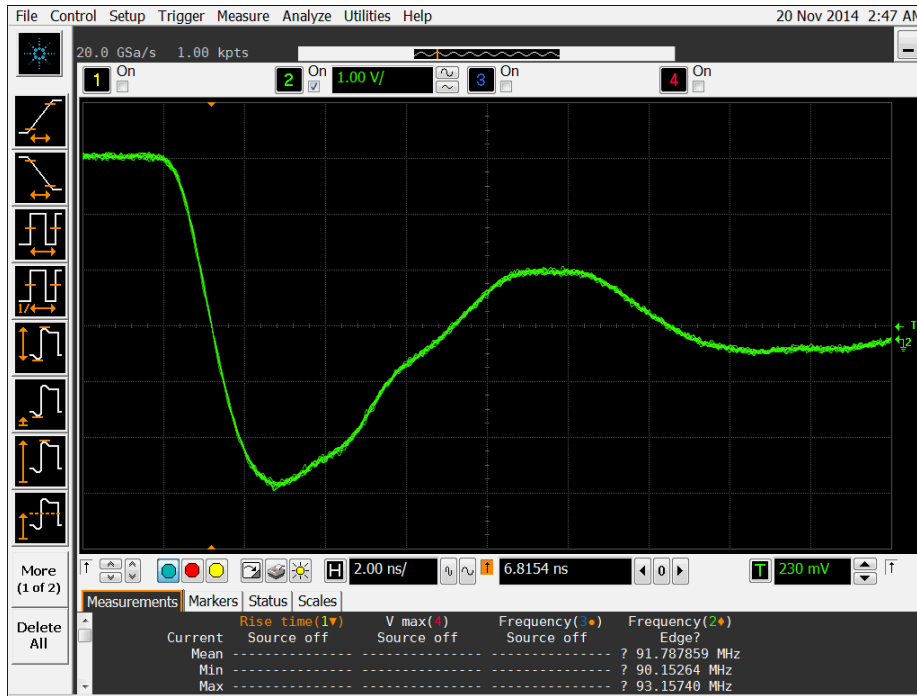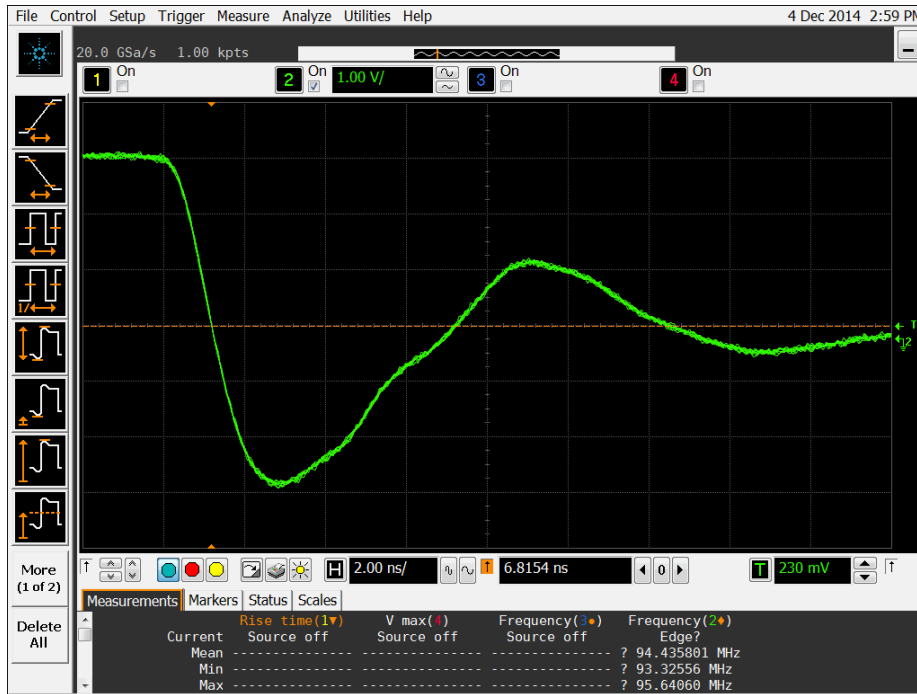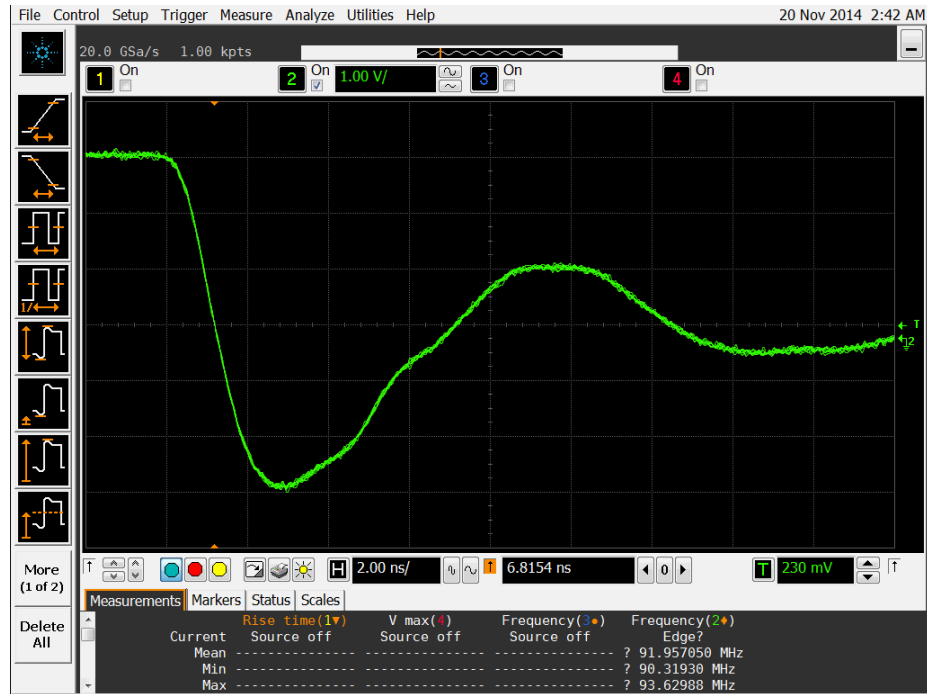abscissa scale is 1 V/div and ordinate scale is 2 ns/div.**

Total Ionizing Dose Test Report No. 15T-RTAX250SL-LG624-D59RQ1

Figure 6a DUT 7959 Pre-Irradiation Rising Edge,
abscissa scale is 1 V/div and ordinate scale is 2 ns/div.

Figure 6b DUT 7959 Post-Annealing Rising Edge,
abscissa scale is 1 V/div and ordinate scale is 2 ns/div.

**Figure 7a DUT 7891 Pre-Irradiation Falling Edge,
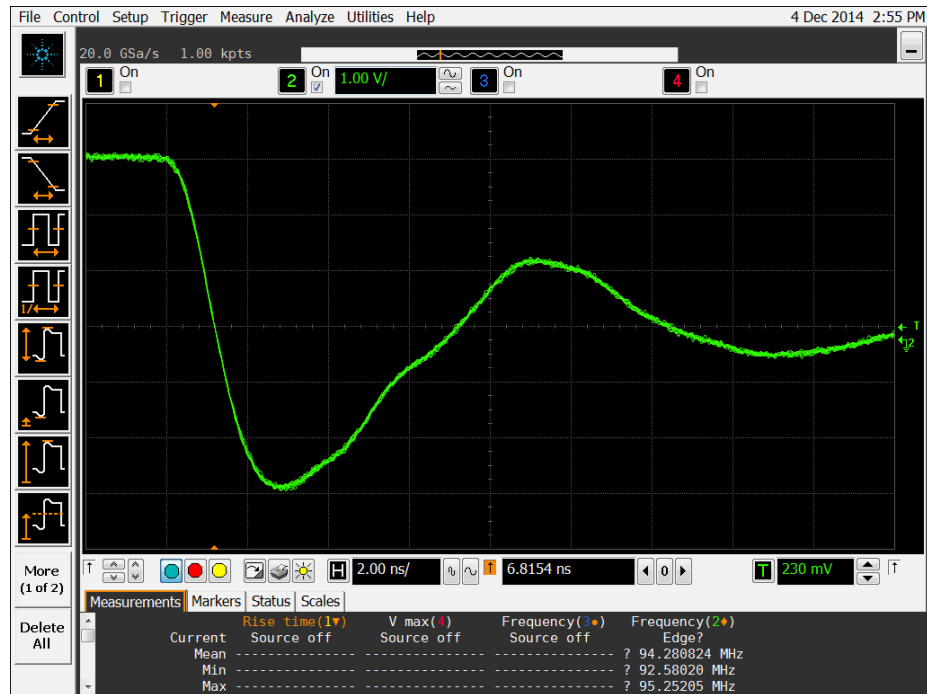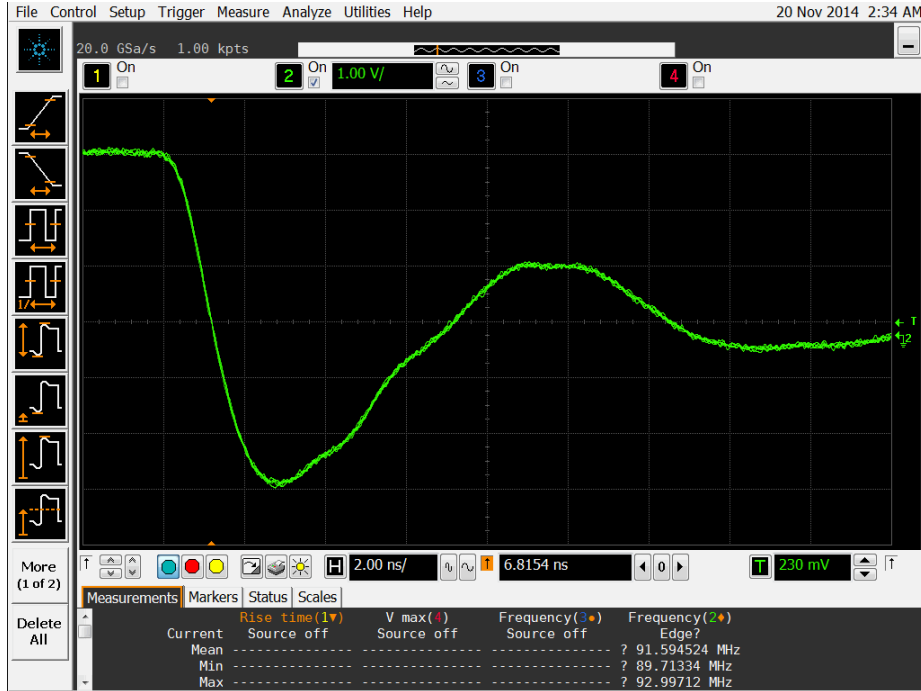abscissa scale is 1 V/div and ordinate scale is 2 ns/div.**

**Figure 7b DUT 7891 Post-Annealing Falling Edge,
abscissa scale is 1 V/div and ordinate scale is 2 ns/div.**

Figure 8a DUT 7904 Pre-Irradiation Falling Edge,
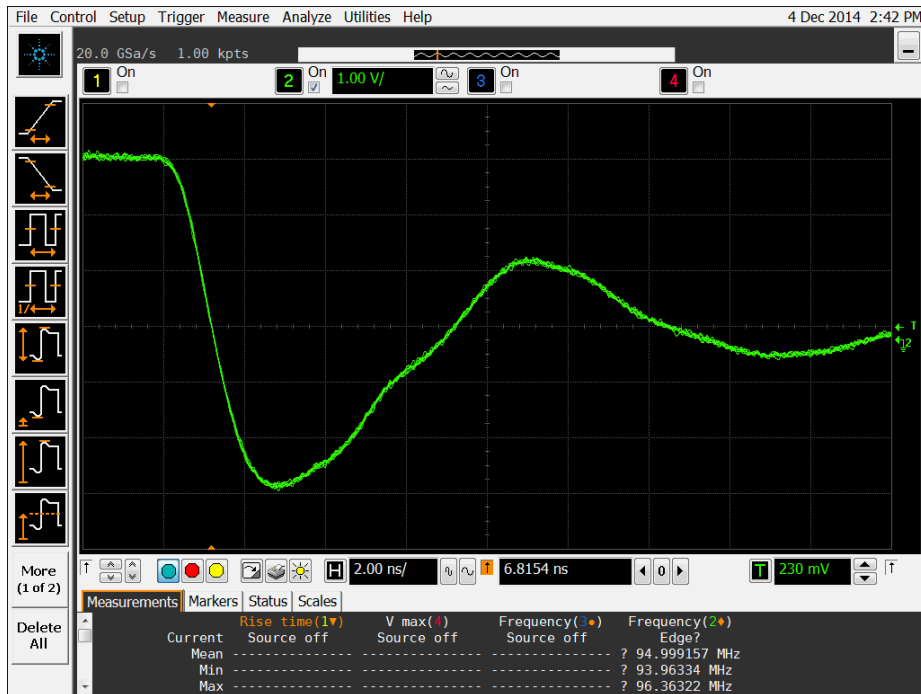**abscissa scale is 1 V/div and ordinate scale is 2 ns/div.**

Figure 8b DUT 7904 Post-Annealing Falling Edge,
abscissa scale is 1 V/div and ordinate scale is 2 ns/div.

File Control Setup Trigger Measure Analyze Utilities Help   20 Nov 2014 2:47 AM

20.0 GSa/s   1.00 kpts

1 On   2 On 1.00 V/   3 On   4 On

More (1 of 2)   2.00 ns/   6.8154 ns   0   230 mV

Delete All

Measurements | Markers | Status | Scales

	Rise time(1▼)	V max(4)	Frequency(3●)	Frequency(2♦)
Current	Source off	Source off	Source off	Edge?
Mean	---------------	---------------	---------------	? 91.787859 MHz
Min	---------------	---------------	---------------	? 90.15264 MHz
Max	---------------	---------------	---------------	? 93.15740 MHz

**Figure 9a DUT 7929 Pre-Irradiation Falling Edge,
abscissa scale is 1 V/div and ordinate scale is 2 ns/div.**

File Control Setup Trigger Measure Analyze Utilities Help   4 Dec 2014 2:59 PM

20.0 GSa/s   1.00 kpts

1 On   2 On 1.00 V/   3 On   4 On

More (1 of 2)   2.00 ns/   6.8154 ns   0   230 mV

Delete All

Measurements | Markers | Status | Scales

	Rise time(1▼)	V max(4)	Frequency(3●)	Frequency(2♦)
Current	Source off	Source off	Source off	Edge?
Mean	---------------	---------------	---------------	? 94.435801 MHz
Min	---------------	---------------	---------------	? 93.32556 MHz
Max	---------------	---------------	---------------	? 95.64060 MHz

**Figure 9b DUT 7929 Post-Annealing Falling Edge,
abscissa scale is 1 V/div and ordinate scale is 2 ns/div.**

**Figure 10a DUT 7937 Pre-Irradiation Falling Edge,
abscissa scale is 1 V/div and ordinate scale is 2 ns/div.**

**Figure 10b DUT 7937 Post-Annealing Falling Edge,
abscissa scale is 1 V/div and ordinate scale is 2 ns/div.**

**Figure 11a DUT 7959 Pre-Irradiation Falling Edge,
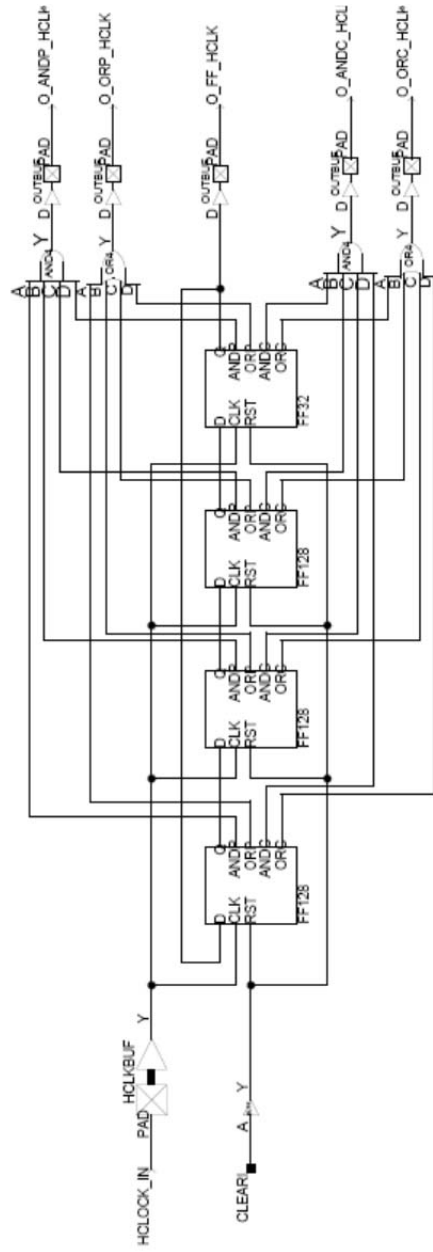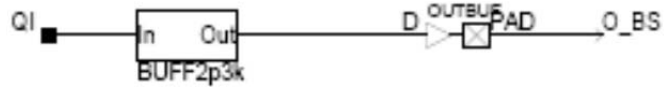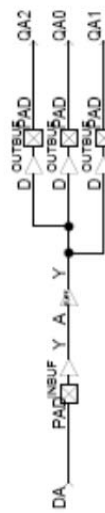abscissa scale is 1 V/div and ordinate scale is 2 ns/div.**

**Figure 11b DUT 7959 Post-Annealing Falling Edge,
abscissa scale is 1 V/div and ordinate scale is 2 ns/div.**
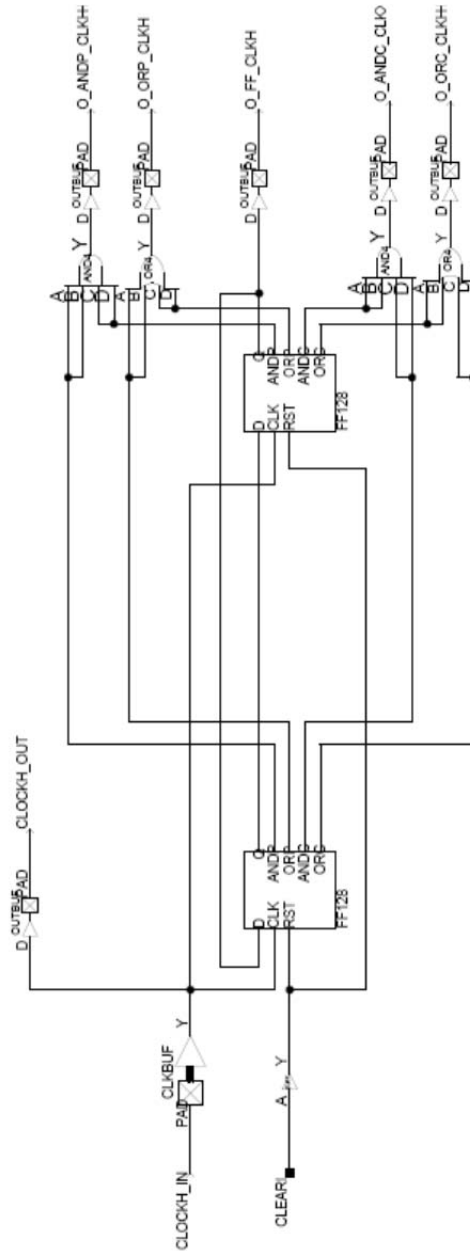
Appendix A: DUT Design Schematics and Verilog Files

QI ■———[In   Out]——————— D ▷OUTBUF [PAD] ———→ O_BS
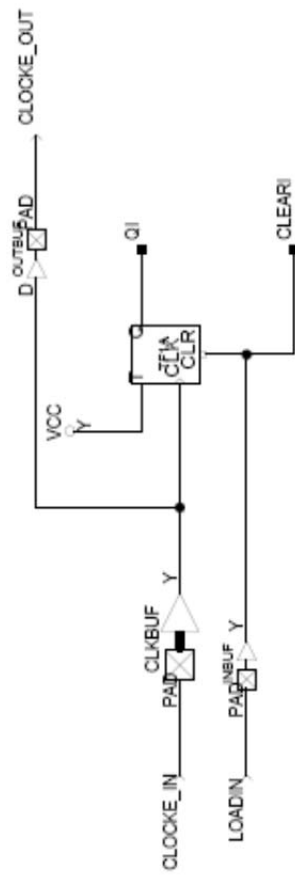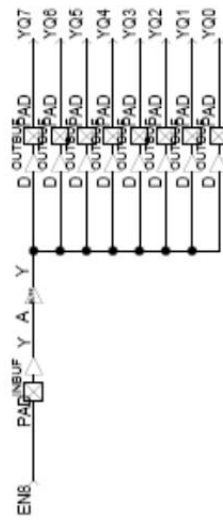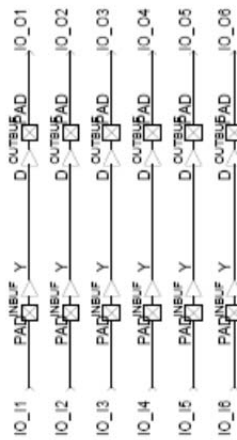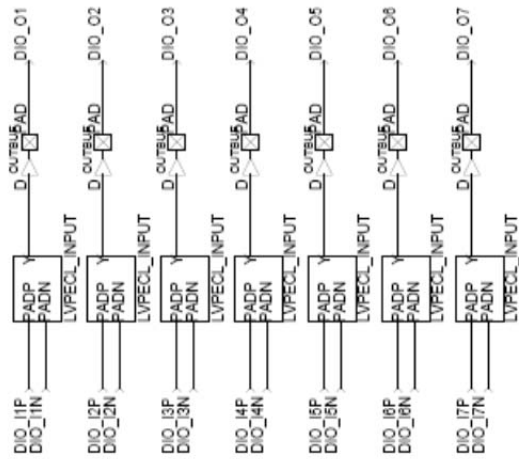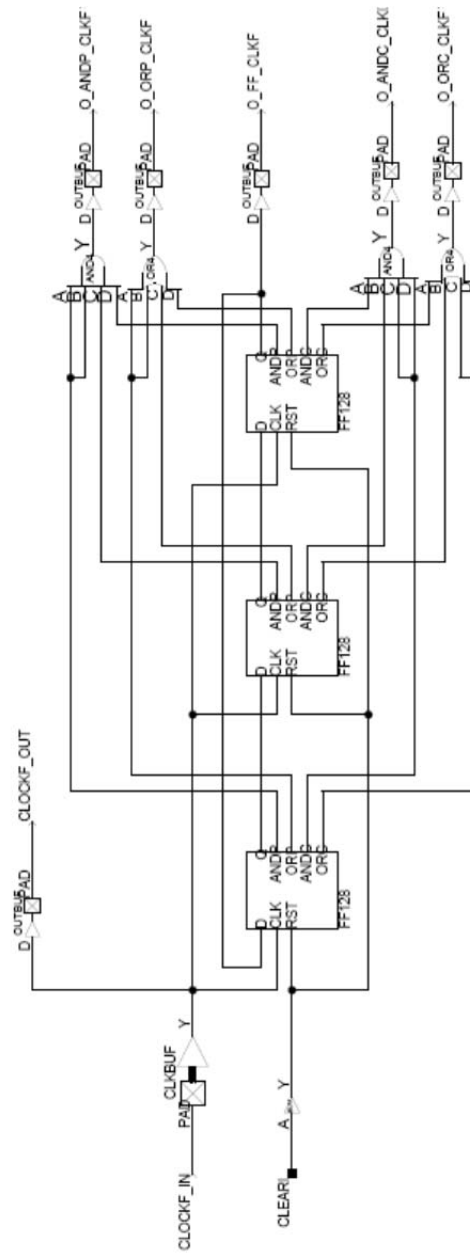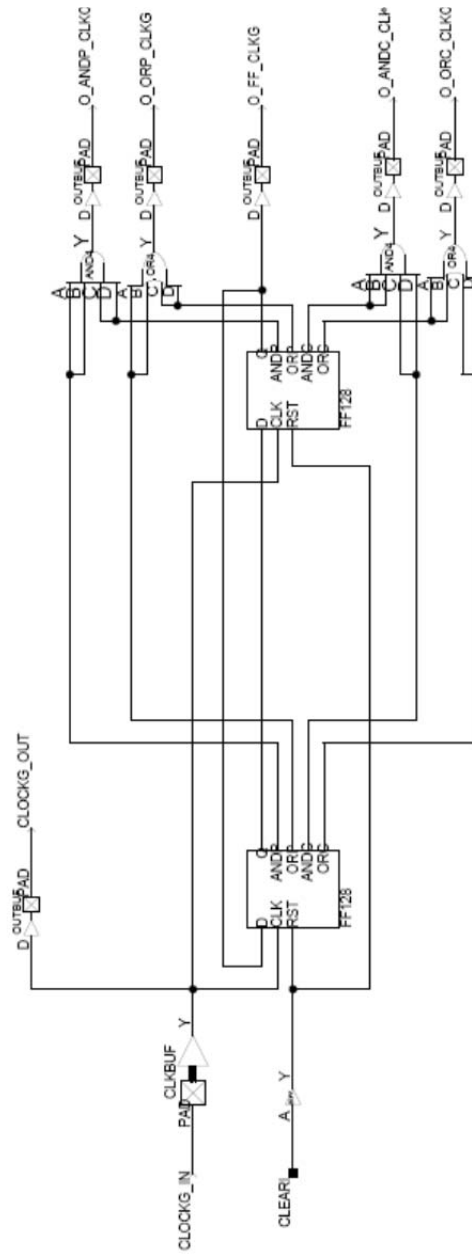
BUFF2p3x

```verilog
// BUFF2p3k.v
`timescale 1 ns/100 ps

module BUFF2p3k (In, Out);

input In;

output Out;

wire x1/*synthesis syn_keep=1 alspreserve=1*/;
wire x2/*synthesis syn_keep=1 alspreserve=1*/;
wire x3/*synthesis syn_keep=1 alspreserve=1*/;
wire x4/*synthesis syn_keep=1 alspreserve=1*/;
wire x5/*synthesis syn_keep=1 alspreserve=1*/;
wire x6/*synthesis syn_keep=1 alspreserve=1*/;
wire x7/*synthesis syn_keep=1 alspreserve=1*/;

BUFF1k buff1k_1 (.In(In), .Out(x1));
BUFF1k buff1k_2 (.In(x1), .Out(x2));
BUFF50 buff3 (.In(x2), .Out(x3));
BUFF50 buff4 (.In(x3), .Out(x4));
BUFF50 buff5 (.In(x4), .Out(x5));
BUFF50 buff6 (.In(x5), .Out(x6));
BUFF50 buff7 (.In(x6), .Out(x7));
BUFF50 buff8 (.In(x7), .Out(Out));

endmodule

// BUFF1k
`timescale 1 ns/100 ps

module BUFF1k (In, Out);

input In;

output Out;

wire x1/*synthesis syn_keep=1 alspreserve=1*/;
wire x2/*synthesis syn_keep=1 alspreserve=1*/;
wire x3/*synthesis syn_keep=1 alspreserve=1*/;
wire x4/*synthesis syn_keep=1 alspreserve=1*/;
wire x5/*synthesis syn_keep=1 alspreserve=1*/;
wire x6/*synthesis syn_keep=1 alspreserve=1*/;
wire x7/*synthesis syn_keep=1 alspreserve=1*/;
wire x8/*synthesis syn_keep=1 alspreserve=1*/;
wire x9/*synthesis syn_keep=1 alspreserve=1*/;
wire x10/*synthesis syn_keep=1 alspreserve=1*/;
wire x11/*synthesis syn_keep=1 alspreserve=1*/;
wire x12/*synthesis syn_keep=1 alspreserve=1*/;
wire x13/*synthesis syn_keep=1 alspreserve=1*/;
wire x14/*synthesis syn_keep=1 alspreserve=1*/;
wire x15/*synthesis syn_keep=1 alspreserve=1*/;
wire x16/*synthesis syn_keep=1 alspreserve=1*/;
wire x17/*synthesis syn_keep=1 alspreserve=1*/;
wire x18/*synthesis syn_keep=1 alspreserve=1*/;
wire x19/*synthesis syn_keep=1 alspreserve=1*/;
```

```verilog
BUFF50 buff1 (.In(In), .Out(x1));
BUFF50 buff2 (.In(x1), .Out(x2));
BUFF50 buff3 (.In(x2), .Out(x3));
BUFF50 buff4 (.In(x3), .Out(x4));
BUFF50 buff5 (.In(x4), .Out(x5));
BUFF50 buff6 (.In(x5), .Out(x6));
BUFF50 buff7 (.In(x6), .Out(x7));
BUFF50 buff8 (.In(x7), .Out(x8));
BUFF50 buff9 (.In(x8), .Out(x9));
BUFF50 buff10 (.In(x9), .Out(x10));

BUFF50 buff11 (.In(x10), .Out(x11));
BUFF50 buff12 (.In(x11), .Out(x12));
BUFF50 buff13 (.In(x12), .Out(x13));
BUFF50 buff14 (.In(x13), .Out(x14));
BUFF50 buff15 (.In(x14), .Out(x15));
BUFF50 buff16 (.In(x15), .Out(x16));
BUFF50 buff17 (.In(x16), .Out(x17));
BUFF50 buff18 (.In(x17), .Out(x18));
BUFF50 buff19 (.In(x18), .Out(x19));
BUFF50 buff20 (.In(x19), .Out(Out));

endmodule

// BUFF50
`timescale 1 ns/100 ps

module BUFF50 (In, Out);

input In;

output Out;

wire x1 /*synthesis syn_keep=1 alspreserve=1*/;
wire x2 /*synthesis syn_keep=1 alspreserve=1*/;
wire x3 /*synthesis syn_keep=1 alspreserve=1*/;
wire x4 /*synthesis syn_keep=1 alspreserve=1*/;
wire x5 /*synthesis syn_keep=1 alspreserve=1*/;
wire x6/*synthesis syn_keep=1 alspreserve=1*/;
wire x7/*synthesis syn_keep=1 alspreserve=1*/;
wire x8/*synthesis syn_keep=1 alspreserve=1*/;
wire x9/*synthesis syn_keep=1 alspreserve=1*/;
wire x10/*synthesis syn_keep=1 alspreserve=1*/;
wire x11/*synthesis syn_keep=1 alspreserve=1*/;
wire x12/*synthesis syn_keep=1 alspreserve=1*/;
wire x13/*synthesis syn_keep=1 alspreserve=1*/;
wire x14/*synthesis syn_keep=1 alspreserve=1*/;
wire x15/*synthesis syn_keep=1 alspreserve=1*/;
wire x16/*synthesis syn_keep=1 alspreserve=1*/;
wire x17/*synthesis syn_keep=1 alspreserve=1*/;
wire x18/*synthesis syn_keep=1 alspreserve=1*/;
wire x19/*synthesis syn_keep=1 alspreserve=1*/;
wire x20/*synthesis syn_keep=1 alspreserve=1*/;
wire x21/*synthesis syn_keep=1 alspreserve=1*/;
wire x22/*synthesis syn_keep=1 alspreserve=1*/;
```

```
wire x23/*synthesis syn_keep=1 alspreserve=1*/;
wire x24/*synthesis syn_keep=1 alspreserve=1*/;
wire x25/*synthesis syn_keep=1 alspreserve=1*/;
wire x26/*synthesis syn_keep=1 alspreserve=1*/;
wire x27/*synthesis syn_keep=1 alspreserve=1*/;
wire x28/*synthesis syn_keep=1 alspreserve=1*/;
wire x29/*synthesis syn_keep=1 alspreserve=1*/;
wire x30/*synthesis syn_keep=1 alspreserve=1*/;
wire x31/*synthesis syn_keep=1 alspreserve=1*/;
wire x32/*synthesis syn_keep=1 alspreserve=1*/;
wire x33/*synthesis syn_keep=1 alspreserve=1*/;
wire x34/*synthesis syn_keep=1 alspreserve=1*/;
wire x35/*synthesis syn_keep=1 alspreserve=1*/;
wire x36/*synthesis syn_keep=1 alspreserve=1*/;
wire x37/*synthesis syn_keep=1 alspreserve=1*/;
wire x38/*synthesis syn_keep=1 alspreserve=1*/;
wire x39/*synthesis syn_keep=1 alspreserve=1*/;
wire x40/*synthesis syn_keep=1 alspreserve=1*/;
wire x41/*synthesis syn_keep=1 alspreserve=1*/;
wire x42/*synthesis syn_keep=1 alspreserve=1*/;
wire x43/*synthesis syn_keep=1 alspreserve=1*/;
wire x44/*synthesis syn_keep=1 alspreserve=1*/;
wire x45/*synthesis syn_keep=1 alspreserve=1*/;
wire x46/*synthesis syn_keep=1 alspreserve=1*/;
wire x47/*synthesis syn_keep=1 alspreserve=1*/;
wire x48/*synthesis syn_keep=1 alspreserve=1*/;
wire x49/*synthesis syn_keep=1 alspreserve=1*/;

BUFF buff1 (.A(In), .Y(x1));
BUFF buff2 (.A(x1), .Y(x2));
BUFF buff3 (.A(x2), .Y(x3));
BUFF buff4 (.A(x3), .Y(x4));
BUFF buff5 (.A(x4), .Y(x5));
BUFF buff6 (.A(x5), .Y(x6));
BUFF buff7 (.A(x6), .Y(x7));
BUFF buff8 (.A(x7), .Y(x8));
BUFF buff9 (.A(x8), .Y(x9));
BUFF buff10 (.A(x9), .Y(x10));

BUFF buff11 (.A(x10), .Y(x11));
BUFF buff12 (.A(x11), .Y(x12));
BUFF buff13 (.A(x12), .Y(x13));
BUFF buff14 (.A(x13), .Y(x14));
BUFF buff15 (.A(x14), .Y(x15));
BUFF buff16 (.A(x15), .Y(x16));
BUFF buff17 (.A(x16), .Y(x17));
BUFF buff18 (.A(x17), .Y(x18));
BUFF buff19 (.A(x18), .Y(x19));
BUFF buff20 (.A(x19), .Y(x20));

BUFF buff21 (.A(x20), .Y(x21));
BUFF buff22 (.A(x21), .Y(x22));
BUFF buff23 (.A(x22), .Y(x23));
BUFF buff24 (.A(x23), .Y(x24));
BUFF buff25 (.A(x24), .Y(x25));
BUFF buff26 (.A(x25), .Y(x26));
```

```
BUFF buff27 (.A(x26), .Y(x27));
BUFF buff28 (.A(x27), .Y(x28));
BUFF buff29 (.A(x28), .Y(x29));
BUFF buff30 (.A(x29), .Y(x30));

BUFF buff31 (.A(x30), .Y(x31));
BUFF buff32 (.A(x31), .Y(x32));
BUFF buff33 (.A(x32), .Y(x33));
BUFF buff34 (.A(x33), .Y(x34));
BUFF buff35 (.A(x34), .Y(x35));
BUFF buff36 (.A(x35), .Y(x36));
BUFF buff37 (.A(x36), .Y(x37));
BUFF buff38 (.A(x37), .Y(x38));
BUFF buff39 (.A(x38), .Y(x39));
BUFF buff40 (.A(x39), .Y(x40));

BUFF buff41 (.A(x40), .Y(x41));
BUFF buff42 (.A(x41), .Y(x42));
BUFF buff43 (.A(x42), .Y(x43));
BUFF buff44 (.A(x43), .Y(x44));
BUFF buff45 (.A(x44), .Y(x45));
BUFF buff46 (.A(x45), .Y(x46));
BUFF buff47 (.A(x46), .Y(x47));
BUFF buff48 (.A(x47), .Y(x48));
BUFF buff49 (.A(x48), .Y(x49));
BUFF buff50 (.A(x49), .Y(Out));

endmodule

// FF128
`timescale 1 ns/100 ps
module FF128 (D, Q, CLK, RST, ANDP, ORP, ANDC, ORC);

input D, CLK, RST;
output Q, ANDP, ORP, ANDC, ORC;

wire x1, x2, x3, Q;
wire andp_a, andp_b, andp_c, andp_d, orp_a, orp_b, orp_c, orp_d;
wire andc_a, andc_b, andc_c, andc_d, orc_a, orc_b, orc_c, orc_d;

FF32 dff_a (.D(D), .Q(x1), .CLK(CLK), .RST(RST), .ANDP(andp_a), .ORP(orp_a),
.ANDC(andc_a), .ORC(orc_a));

FF32 dff_b (.D(x1), .Q(x2), .CLK(CLK), .RST(RST), .ANDP(andp_b), .ORP(orp_b),
.ANDC(andc_b), .ORC(orc_b));

FF32 dff_c (.D(x2), .Q(x3), .CLK(CLK), .RST(RST), .ANDP(andp_c), .ORP(orp_c),
.ANDC(andc_c), .ORC(orc_c));

FF32 dff_d (.D(x3), .Q(Q), .CLK(CLK), .RST(RST), .ANDP(andp_d), .ORP(orp_d),
.ANDC(andc_d), .ORC(orc_d));

AND4 and4p (.A(andp_a), .B(andp_b), .C(andp_c), .D(andp_d), .Y(ANDP));
OR4 or4p (.A(orp_a), .B(orp_b), .C(orp_c), .D(orp_d), .Y(ORP));

AND4 and4c (.A(andc_a), .B(andc_b), .C(andc_c), .D(andc_d), .Y(ANDC));
```

```verilog
OR4 or4c (.A(orc_a), .B(orc_b), .C(orc_c), .D(orc_d), .Y(ORC));

endmodule

// FF32
`timescale 1 ns/100 ps
module FF32 (D, Q, CLK, RST, ANDP, ORP, ANDC, ORC);

input D, CLK, RST;
output Q, ANDP, ORP, ANDC, ORC;

wire x1, x2, x3, Q;
wire andp_a, andp_b, andp_c, andp_d, orp_a, orp_b, orp_c, orp_d;
wire andc_a, andc_b, andc_c, andc_d, orc_a, orc_b, orc_c, orc_d;

FF8 dff_a (.D(D), .Q(x1), .CLK(CLK), .RST(RST), .ANDP(andp_a), .ORP(orp_a),
.ANDC(andc_a), .ORC(orc_a));

FF8 dff_b (.D(x1), .Q(x2), .CLK(CLK), .RST(RST), .ANDP(andp_b), .ORP(orp_b),
.ANDC(andc_b), .ORC(orc_b));

FF8 dff_c (.D(x2), .Q(x3), .CLK(CLK), .RST(RST), .ANDP(andp_c), .ORP(orp_c),
.ANDC(andc_c), .ORC(orc_c));

FF8 dff_d (.D(x3), .Q(Q), .CLK(CLK), .RST(RST), .ANDP(andp_d), .ORP(orp_d),
.ANDC(andc_d), .ORC(orc_d));

AND4 and4p (.A(andp_a), .B(andp_b), .C(andp_c), .D(andp_d), .Y(ANDP));
OR4 or4p (.A(orp_a), .B(orp_b), .C(orp_c), .D(orp_d), .Y(ORP));

AND4 and4c (.A(andc_a), .B(andc_b), .C(andc_c), .D(andc_d), .Y(ANDC));
OR4 or4c (.A(orc_a), .B(orc_b), .C(orc_c), .D(orc_d), .Y(ORC));

endmodule

// FF8
`timescale 1 ns/100 ps

module FF8 (D, Q, CLK, RST, ANDP, ORP, ANDC, ORC);

input D, CLK, RST;
output Q, ANDP, ORP, ANDC, ORC;

wire x1, x2, x3, x4, x5, x6, x7;

DFC1B dff1 (.D(D), .Q(x1), .CLK(CLK), .CLR(RST));
DFP1B dff2 (.D(x1), .Q(x2), .CLK(CLK), .PRE(RST));
DFC1B dff3 (.D(x2), .Q(x3), .CLK(CLK), .CLR(RST));
DFP1B dff4 (.D(x3), .Q(x4), .CLK(CLK), .PRE(RST));
DFC1B dff5 (.D(x4), .Q(x5), .CLK(CLK), .CLR(RST));
DFP1B dff6 (.D(x5), .Q(x6), .CLK(CLK), .PRE(RST));
DFC1B dff7 (.D(x6), .Q(x7), .CLK(CLK), .CLR(RST));
DFP1B dff8 (.D(x7), .Q(Q), .CLK(CLK), .PRE(RST));

AND4 and4p (.A(x2), .B(x4), .C(x6), .D(Q), .Y(ANDP));
OR4 or4p (.A(x2), .B(x4), .C(x6), .D(Q), .Y(ORP));
```

```verilog
AND4 and4c (.A(x1), .B(x3), .C(x5), .D(x7), .Y(ANDC));
OR4 or4c (.A(x1), .B(x3), .C(x5), .D(x7), .Y(ORC));

endmodule

// Top_RAM_Module.v
`timescale 1 ns/100 ps

module Top_RAM_Module(Psel0, Psel1, RC_en, RC_clr, RC_clk, Write, Read, Wclk, Rclk,
          Q_RAM);
input Psel0, Psel1, RC_en, RC_clr, RC_clk, Write, Read, Wclk, Rclk;
output [5:0] Q_RAM;

wire Gnd, Vcc;
wire mx0, mx1;
wire [12:0] rc;
wire [3:0] dec;
wire y_0w, y_0r, y_1w, y_1r, y_2w, y_2r, y_3w, y_3r;
// y_4w, y_4r, y_5w, y_5r, y_6w, y_6r, y_7w, y_7r;
wire [5:0] DIN;
wire [5:0] Q_b0;
wire [5:0] Q_b1;
wire [5:0] Q_b2;
wire [5:0] Q_b3;
//wire [5:0] Q_b4;
//wire [5:0] Q_b5;
//wire [5:0] Q_b6;
//wire [5:0] Q_b7;

GND gnd_0(.Y(Gnd));
VCC vcc_0(.Y(Vcc));

mux_2x1 mux_0(.Data0_port(Gnd), .Data1_port(Vcc), .Sel0(Psel0), .Result(mx0));
mux_2x1 mux_1(.Data0_port(Gnd), .Data1_port(Vcc), .Sel0(Psel1), .Result(mx1));

counter_13 counter_0(.Enable(RC_en), .Aclr(RC_clr), .Clock(RC_clk), .Q(rc));

decoder_2to4 decoder_0(.Data0(rc[11]), .Data1(rc[12]), .Eq(dec));

NAND2 nand_0w(.A(dec[0]), .B(Write), .Y(y_0w));
NAND2 nand_0r(.A(dec[0]), .B(Read), .Y(y_0r));

ram_2048x6 ram_blk0(.Data(DIN), .Q(Q_b0), .WAddress(rc[10:0]), .RAddress(rc[10:0]),
          .WE(y_0w), .RE(y_0r), .WClock(Wclk), .RClock(Rclk));

assign DIN[0]=mx0, DIN[1]=mx1, DIN[2]=mx0, DIN[3]=mx1, DIN[4]=mx0, DIN[5]=mx1;

NAND2 nand_1w(.A(dec[1]), .B(Write), .Y(y_1w));
NAND2 nand_1r(.A(dec[1]), .B(Read), .Y(y_1r));

ram_2048x6 ram_blk1(.Data(DIN), .Q(Q_b1), .WAddress(rc[10:0]), .RAddress(rc[10:0]),
          .WE(y_1w), .RE(y_1r), .WClock(Wclk), .RClock(Rclk));

NAND2 nand_2w(.A(dec[2]), .B(Write), .Y(y_2w));
NAND2 nand_2r(.A(dec[2]), .B(Read), .Y(y_2r));
```

```
ram_2048x6 ram_blk2(.Data(DIN),
           .Q(Q_b2), .WAddress(rc[10:0]), .RAddress(rc[10:0]),
           .WE(y_2w), .RE(y_2r), .WClock(Wclk), .RClock(Rclk));

NAND2 nand_3w(.A(dec[3]), .B(Write), .Y(y_3w));
NAND2 nand_3r(.A(dec[3]), .B(Read), .Y(y_3r));

ram_2048x6 ram_blk3(.Data(DIN),
           .Q(Q_b3), .WAddress(rc[10:0]), .RAddress(rc[10:0]),
           .WE(y_3w), .RE(y_3r), .WClock(Wclk), .RClock(Rclk));

/* NAND2 nand_4w(.A(dec[4]), .B(Write), .Y(y_4w));
NAND2 nand_4r(.A(dec[4]), .B(Read), .Y(y_4r));

ram_2048x3 ram_blk4(.Data(DIN),
           .Q(Q_b4), .WAddress(rc[10:0]), .RAddress(rc[10:0]),
           .WE(y_4w), .RE(y_4r), .WClock(Wclk), .RClock(Rclk));

NAND2 nand_5w(.A(dec[5]), .B(Write), .Y(y_5w));
NAND2 nand_5r(.A(dec[5]), .B(Read), .Y(y_5r));

ram_2048x3 ram_blk5(.Data(DIN),
           .Q(Q_b5), .WAddress(rc[10:0]), .RAddress(rc[10:0]),
           .WE(y_5w), .RE(y_5r), .WClock(Wclk), .RClock(Rclk));

NAND2 nand_6w(.A(dec[6]), .B(Write), .Y(y_6w));
NAND2 nand_6r(.A(dec[6]), .B(Read), .Y(y_6r));

ram_2048x3 ram_blk6(.Data(DIN),
           .Q(Q_b6), .WAddress(rc[10:0]), .RAddress(rc[10:0]),
           .WE(y_6w), .RE(y_6r), .WClock(Wclk), .RClock(Rclk));

NAND2 nand_7w(.A(dec[7]), .B(Write), .Y(y_7w));
NAND2 nand_7r(.A(dec[7]), .B(Read), .Y(y_7r));

ram_2048x3 ram_blk7(.Data(DIN),
           .Q(Q_b7), .WAddress(rc[10:0]), .RAddress(rc[10:0]),
           .WE(y_7w), .RE(y_7r), .WClock(Wclk), .RClock(Rclk)); */

mux_6x4 mux_6x4_0(.Data0_port(Q_b0), .Data1_port(Q_b1), .Data2_port(Q_b2),
           .Data3_port(Q_b3), .Sel0(rc[11]),
           .Sel1(rc[12]), .Result(Q_RAM));

endmodule
```

`timescale 1 ns/100 ps
// Version: 6.0 SP3 6.0.30.3

module mux_2x1(Data0_port,Data1_port,Sel0,Result);
input Data0_port, Data1_port, Sel0;
output Result;

   MX2 MX2_Result(.A(Data0_port), .B(Data1_port), .S(Sel0), .Y(
      Result));

endmodule

`timescale 1 ns/100 ps
// Version: 6.2 SP2 6.2.52.7

module counter_13(Enable,Aclr,Clock,Q);
input Enable, Aclr, Clock;
output [12:0] Q;

   wire ClrAux_0_net, ClrAux_7_net, MX2_1_Y, MX2_7_Y, MX2_4_Y,
      CM8_0_Y, MX2_10_Y, MX2_9_Y, MX2_3_Y, MX2_5_Y, MX2_6_Y,
      MX2_0_Y, MX2_8_Y, MX2_2_Y, MX2_11_Y, VCC, GND;

   VCC VCC_1_net(.Y(VCC));
   GND GND_1_net(.Y(GND));
   DFC1D DFC1D_Q_7_inst(.D(MX2_1_Y), .CLK(Q[6]), .CLR(
      ClrAux_7_net), .Q(Q[7]));
   DFC1D DFC1D_Q_1_inst(.D(MX2_7_Y), .CLK(Q[0]), .CLR(
      ClrAux_0_net), .Q(Q[1]));
   BUFF BUFF_ClrAux_0_inst(.A(Aclr), .Y(ClrAux_0_net));
   MX2 MX2_9(.A(VCC), .B(GND), .S(Q[5]), .Y(MX2_9_Y));
   DFC1D DFC1D_Q_2_inst(.D(MX2_6_Y), .CLK(Q[1]), .CLR(
      ClrAux_0_net), .Q(Q[2]));
   MX2 MX2_0(.A(VCC), .B(GND), .S(Q[8]), .Y(MX2_0_Y));
   DFC1D DFC1D_Q_12_inst(.D(MX2_4_Y), .CLK(Q[11]), .CLR(
      ClrAux_7_net), .Q(Q[12]));
   DFC1D DFC1D_Q_3_inst(.D(MX2_11_Y), .CLK(Q[2]), .CLR(
      ClrAux_0_net), .Q(Q[3]));
   DFC1D DFC1D_Q_4_inst(.D(MX2_5_Y), .CLK(Q[3]), .CLR(
      ClrAux_0_net), .Q(Q[4]));
   CM8 CM8_0(.D0(GND), .D1(VCC), .D2(VCC), .D3(GND), .S00(Q[0]),
      .S01(VCC), .S10(Enable), .S11(GND), .Y(CM8_0_Y));
   MX2 MX2_11(.A(VCC), .B(GND), .S(Q[3]), .Y(MX2_11_Y));
   DFC1B DFC1B_Q_0_inst(.D(CM8_0_Y), .CLK(Clock), .CLR(
      ClrAux_0_net), .Q(Q[0]));
   MX2 MX2_6(.A(VCC), .B(GND), .S(Q[2]), .Y(MX2_6_Y));
   MX2 MX2_3(.A(VCC), .B(GND), .S(Q[10]), .Y(MX2_3_Y));
   DFC1D DFC1D_Q_11_inst(.D(MX2_10_Y), .CLK(Q[10]), .CLR(
      ClrAux_7_net), .Q(Q[11]));
   MX2 MX2_10(.A(VCC), .B(GND), .S(Q[11]), .Y(MX2_10_Y));
   BUFF BUFF_ClrAux_7_inst(.A(Aclr), .Y(ClrAux_7_net));
   MX2 MX2_4(.A(VCC), .B(GND), .S(Q[12]), .Y(MX2_4_Y));
   DFC1D DFC1D_Q_5_inst(.D(MX2_9_Y), .CLK(Q[4]), .CLR(

```
      ClrAux_0_net), .Q(Q[5]));
   DFC1D DFC1D_Q_9_inst(.D(MX2_8_Y), .CLK(Q[8]), .CLR(
      ClrAux_7_net), .Q(Q[9]));
   MX2 MX2_5(.A(VCC), .B(GND), .S(Q[4]), .Y(MX2_5_Y));
   MX2 MX2_8(.A(VCC), .B(GND), .S(Q[9]), .Y(MX2_8_Y));
   DFC1D DFC1D_Q_8_inst(.D(MX2_0_Y), .CLK(Q[7]), .CLR(
      ClrAux_7_net), .Q(Q[8]));
   MX2 MX2_2(.A(VCC), .B(GND), .S(Q[6]), .Y(MX2_2_Y));
   MX2 MX2_7(.A(VCC), .B(GND), .S(Q[1]), .Y(MX2_7_Y));
   MX2 MX2_1(.A(VCC), .B(GND), .S(Q[7]), .Y(MX2_1_Y));
   DFC1D DFC1D_Q_6_inst(.D(MX2_2_Y), .CLK(Q[5]), .CLR(
      ClrAux_0_net), .Q(Q[6]));
   DFC1D DFC1D_Q_10_inst(.D(MX2_3_Y), .CLK(Q[9]), .CLR(
      ClrAux_7_net), .Q(Q[10]));

endmodule

`timescale 1 ns/100 ps
// Version: 6.2 SP2 6.2.52.7


module decoder_2to4(Data0,Data1,Eq);
input Data0, Data1;
output [3:0] Eq;


   AND2A AND2A_Eq_1_inst(.A(Data1), .B(Data0), .Y(Eq[1]));
   AND2 AND2_Eq_3_inst(.A(Data0), .B(Data1), .Y(Eq[3]));
   AND2A AND2A_Eq_2_inst(.A(Data0), .B(Data1), .Y(Eq[2]));
   AND2B AND2B_Eq_0_inst(.A(Data0), .B(Data1), .Y(Eq[0]));

endmodule

`timescale 1 ns/100 ps
// Version: 6.2 SP2 6.2.52.7


module ram_2048x6(Data,Q,WAddress,RAddress,WE,RE,WClock,RClock);
input [5:0] Data;
output [5:0] Q;
input [10:0] WAddress, RAddress;
input WE, RE, WClock, RClock;

   wire WEP, REP, VCC, GND;

   VCC VCC_1_net(.Y(VCC));
   GND GND_1_net(.Y(GND));
   RAM64K36P ram_2048x6_R0C2(.WCLK(WClock), .RCLK(RClock),
      .DEPTH0(GND), .DEPTH1(GND), .DEPTH2(GND), .DEPTH3(GND),
      .WEN(WEP), .WW0(VCC), .WW1(GND), .WW2(GND), .WRAD0(
      WAddress[0]), .WRAD1(WAddress[1]), .WRAD2(WAddress[2]),
      .WRAD3(WAddress[3]), .WRAD4(WAddress[4]), .WRAD5(
      WAddress[5]), .WRAD6(WAddress[6]), .WRAD7(WAddress[7]),
      .WRAD8(WAddress[8]), .WRAD9(WAddress[9]), .WRAD10(
      WAddress[10]), .WRAD11(GND), .WRAD12(GND), .WRAD13(GND),
      .WRAD14(GND), .WRAD15(GND), .WD0(Data[4]), .WD1(Data[5]),
```

```
      .WD2(GND), .WD3(GND), .WD4(GND), .WD5(GND), .WD6(GND),
      .WD7(GND), .WD8(GND), .WD9(GND), .WD10(GND), .WD11(GND),
      .WD12(GND), .WD13(GND), .WD14(GND), .WD15(GND), .WD16(GND)
      , .WD17(GND), .WD18(GND), .WD19(GND), .WD20(GND), .WD21(
      GND), .WD22(GND), .WD23(GND), .WD24(GND), .WD25(GND),
      .WD26(GND), .WD27(GND), .WD28(GND), .WD29(GND), .WD30(GND)
      , .WD31(GND), .WD32(GND), .WD33(GND), .WD34(GND), .WD35(
      GND), .REN(REP), .RW0(VCC), .RW1(GND), .RW2(GND), .RDAD0(
      RAddress[0]), .RDAD1(RAddress[1]), .RDAD2(RAddress[2]),
      .RDAD3(RAddress[3]), .RDAD4(RAddress[4]), .RDAD5(
      RAddress[5]), .RDAD6(RAddress[6]), .RDAD7(RAddress[7]),
      .RDAD8(RAddress[8]), .RDAD9(RAddress[9]), .RDAD10(
      RAddress[10]), .RDAD11(GND), .RDAD12(GND), .RDAD13(GND),
      .RDAD14(GND), .RDAD15(GND), .RD0(Q[4]), .RD1(Q[5]), .RD2()
      , .RD3(), .RD4(), .RD5(), .RD6(), .RD7(), .RD8(), .RD9(),
      .RD10(), .RD11(), .RD12(), .RD13(), .RD14(), .RD15(),
      .RD16(), .RD17(), .RD18(), .RD19(), .RD20(), .RD21(),
      .RD22(), .RD23(), .RD24(), .RD25(), .RD26(), .RD27(),
      .RD28(), .RD29(), .RD30(), .RD31(), .RD32(), .RD33(),
      .RD34(), .RD35());
   INV REBUBBLE(.A(RE), .Y(REP));
   INV WEBUBBLE(.A(WE), .Y(WEP));
   RAM64K36P ram_2048x6_R0C1(.WCLK(WClock), .RCLK(RClock),
      .DEPTH0(GND), .DEPTH1(GND), .DEPTH2(GND), .DEPTH3(GND),
      .WEN(WEP), .WW0(VCC), .WW1(GND), .WW2(GND), .WRAD0(
      WAddress[0]), .WRAD1(WAddress[1]), .WRAD2(WAddress[2]),
      .WRAD3(WAddress[3]), .WRAD4(WAddress[4]), .WRAD5(
      WAddress[5]), .WRAD6(WAddress[6]), .WRAD7(WAddress[7]),
      .WRAD8(WAddress[8]), .WRAD9(WAddress[9]), .WRAD10(
      WAddress[10]), .WRAD11(GND), .WRAD12(GND), .WRAD13(GND),
      .WRAD14(GND), .WRAD15(GND), .WD0(Data[2]), .WD1(Data[3]),
      .WD2(GND), .WD3(GND), .WD4(GND), .WD5(GND), .WD6(GND),
      .WD7(GND), .WD8(GND), .WD9(GND), .WD10(GND), .WD11(GND),
      .WD12(GND), .WD13(GND), .WD14(GND), .WD15(GND), .WD16(GND)
      , .WD17(GND), .WD18(GND), .WD19(GND), .WD20(GND), .WD21(
      GND), .WD22(GND), .WD23(GND), .WD24(GND), .WD25(GND),
      .WD26(GND), .WD27(GND), .WD28(GND), .WD29(GND), .WD30(GND)
      , .WD31(GND), .WD32(GND), .WD33(GND), .WD34(GND), .WD35(
      GND), .REN(REP), .RW0(VCC), .RW1(GND), .RW2(GND), .RDAD0(
      RAddress[0]), .RDAD1(RAddress[1]), .RDAD2(RAddress[2]),
      .RDAD3(RAddress[3]), .RDAD4(RAddress[4]), .RDAD5(
      RAddress[5]), .RDAD6(RAddress[6]), .RDAD7(RAddress[7]),
      .RDAD8(RAddress[8]), .RDAD9(RAddress[9]), .RDAD10(
      RAddress[10]), .RDAD11(GND), .RDAD12(GND), .RDAD13(GND),
      .RDAD14(GND), .RDAD15(GND), .RD0(Q[2]), .RD1(Q[3]), .RD2()
      , .RD3(), .RD4(), .RD5(), .RD6(), .RD7(), .RD8(), .RD9(),
      .RD10(), .RD11(), .RD12(), .RD13(), .RD14(), .RD15(),
      .RD16(), .RD17(), .RD18(), .RD19(), .RD20(), .RD21(),
      .RD22(), .RD23(), .RD24(), .RD25(), .RD26(), .RD27(),
      .RD28(), .RD29(), .RD30(), .RD31(), .RD32(), .RD33(),
      .RD34(), .RD35());
   RAM64K36P ram_2048x6_R0C0(.WCLK(WClock), .RCLK(RClock),
      .DEPTH0(GND), .DEPTH1(GND), .DEPTH2(GND), .DEPTH3(GND),
      .WEN(WEP), .WW0(VCC), .WW1(GND), .WW2(GND), .WRAD0(
      WAddress[0]), .WRAD1(WAddress[1]), .WRAD2(WAddress[2]),
      .WRAD3(WAddress[3]), .WRAD4(WAddress[4]), .WRAD5(
```

WAddress[5]), .WRAD6(WAddress[6]), .WRAD7(WAddress[7]),
.WRAD8(WAddress[8]), .WRAD9(WAddress[9]), .WRAD10(
WAddress[10]), .WRAD11(GND), .WRAD12(GND), .WRAD13(GND),
.WRAD14(GND), .WRAD15(GND), .WD0(Data[0]), .WD1(Data[1]),
.WD2(GND), .WD3(GND), .WD4(GND), .WD5(GND), .WD6(GND),
.WD7(GND), .WD8(GND), .WD9(GND), .WD10(GND), .WD11(GND),
.WD12(GND), .WD13(GND), .WD14(GND), .WD15(GND), .WD16(GND)
, .WD17(GND), .WD18(GND), .WD19(GND), .WD20(GND), .WD21(
GND), .WD22(GND), .WD23(GND), .WD24(GND), .WD25(GND),
.WD26(GND), .WD27(GND), .WD28(GND), .WD29(GND), .WD30(GND)
, .WD31(GND), .WD32(GND), .WD33(GND), .WD34(GND), .WD35(
GND), .REN(REP), .RW0(VCC), .RW1(GND), .RW2(GND), .RDAD0(
RAddress[0]), .RDAD1(RAddress[1]), .RDAD2(RAddress[2]),
.RDAD3(RAddress[3]), .RDAD4(RAddress[4]), .RDAD5(
RAddress[5]), .RDAD6(RAddress[6]), .RDAD7(RAddress[7]),
.RDAD8(RAddress[8]), .RDAD9(RAddress[9]), .RDAD10(
RAddress[10]), .RDAD11(GND), .RDAD12(GND), .RDAD13(GND),
.RDAD14(GND), .RDAD15(GND), .RD0(Q[0]), .RD1(Q[1]), .RD2()
, .RD3(), .RD4(), .RD5(), .RD6(), .RD7(), .RD8(), .RD9(),
.RD10(), .RD11(), .RD12(), .RD13(), .RD14(), .RD15(),
.RD16(), .RD17(), .RD18(), .RD19(), .RD20(), .RD21(),
.RD22(), .RD23(), .RD24(), .RD25(), .RD26(), .RD27(),
.RD28(), .RD29(), .RD30(), .RD31(), .RD32(), .RD33(),
.RD34(), .RD35());

endmodule

`timescale 1 ns/100 ps
// Version: 6.2 SP2 6.2.52.7

module mux_6x4(Data0_port,Data1_port,Data2_port,Data3_port,Sel0,
    Sel1,Result);
input [5:0] Data0_port, Data1_port, Data2_port, Data3_port;
input Sel0, Sel1;
output [5:0] Result;

  MX4 MX4_Result_0_inst(.D0(Data0_port[0]), .D1(Data1_port[0]),
    .D2(Data2_port[0]), .D3(Data3_port[0]), .S0(Sel0), .S1(
    Sel1), .Y(Result[0]));
  MX4 MX4_Result_2_inst(.D0(Data0_port[2]), .D1(Data1_port[2]),
    .D2(Data2_port[2]), .D3(Data3_port[2]), .S0(Sel0), .S1(
    Sel1), .Y(Result[2]));
  MX4 MX4_Result_5_inst(.D0(Data0_port[5]), .D1(Data1_port[5]),
    .D2(Data2_port[5]), .D3(Data3_port[5]), .S0(Sel0), .S1(
    Sel1), .Y(Result[5]));
  MX4 MX4_Result_1_inst(.D0(Data0_port[1]), .D1(Data1_port[1]),
    .D2(Data2_port[1]), .D3(Data3_port[1]), .S0(Sel0), .S1(
    Sel1), .Y(Result[1]));
  MX4 MX4_Result_4_inst(.D0(Data0_port[4]), .D1(Data1_port[4]),
    .D2(Data2_port[4]), .D3(Data3_port[4]), .S0(Sel0), .S1(
    Sel1), .Y(Result[4]));
  MX4 MX4_Result_3_inst(.D0(Data0_port[3]), .D1(Data1_port[3]),
    .D2(Data2_port[3]), .D3(Data3_port[3]), .S0(Sel0), .S1(
    Sel1), .Y(Result[3]));

endmodule