



## TOTAL IONIZING DOSE TEST REPORT

No. 08T-RTAX2000SL-D3HY31

November 18, 2008

J.J. Wang

(650) 318-4576

[jih-jong.wang@actel.com](mailto:jih-jong.wang@actel.com)

### I. SUMMARY TABLE

Parameter	Tolerance
1. Gross Functionality	Passed 300 krad(SiO <sub>2</sub> )
2. Power Supply Current (I <sub>CCA</sub> /I <sub>CCI</sub> )	Passed 300 krad(SiO <sub>2</sub> )
3. Input Threshold (V <sub>TIL</sub> /V <sub>IH</sub> )	Passed 300 krad(SiO <sub>2</sub> )
4. Output Drive (V <sub>OL</sub> /V <sub>OH</sub> )	Passed 300 krad(SiO <sub>2</sub> )
5. Propagation Delay	Passed 300 krad(SiO <sub>2</sub> ) for ±10% degradation criterion
6. Transition Time	Passed 300 krad(SiO <sub>2</sub> )

### II. TOTAL IONIZING DOSE (TID) TESTING

This testing is designed on the base of an extensive database (see TID data of antifuse-based FPGA in <http://www.actel.com/> and <http://www.klabs.org/>) 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 each input and most of the output is grounded through a 100k-ohm resistor; during annealing each input or output is tied to the ground or V<sub>CCI</sub> with a 2.7 k ohm resistor. Appendix A contains the schematics of the irradiation-bias circuit.

Table 1 DUT and Irradiation Parameters

Part Number	RTAX2000SL
Package	CQFP352
Foundry	United Microelectronics Corp.
Technology	0.15 μm CMOS
DUT Design	TOP_AX2000S_TID
Die Lot Number	D3HY31
Quantity Tested	5
Serial Number	200 krad: 7756, 7776 300 krad: 7734, 7741, 7745
Radiation Facility	Defense Microelectronics Activity
Radiation Source	Co-60
Dose Rate (±5%)	7.5 krad(SiO <sub>2</sub> )/min
Irradiation Temperature	Room
Irradiation and Measurement Bias (V <sub>CCI</sub> /V <sub>CCA</sub> )	Static at 3.3 V/1.5 V
IO Configuration	Single ended: LVTTTL Differential pair: LVPECL

B. Test Method

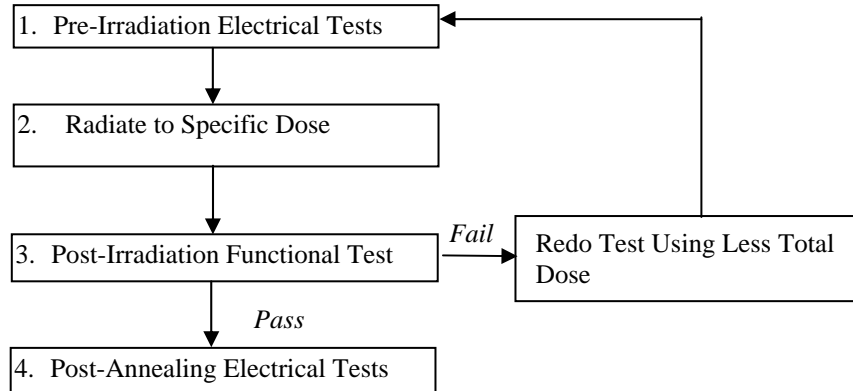


Figure 1 Parametric test flow chart

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

The accelerated aging, or rebound test mentioned in TM1019 is unnecessary because there is no adverse time-dependent effect (TDE) in Actel products manufactured by deep sub-micron CMOS technologies. Elevated temperature annealing basically reduces the effects originating from radiation-induced leakage currents. As indicated by test data in the following sections, the predominant radiation effects in RTAX2000S are due to radiation-induced leakage currents.

Room temperature annealing is performed in this test; the duration is approximately 10 days.

### C. Design and Parametric Measurements

The DUT uses a high utilization generic design (TOP\_AX2000S\_TID) to evaluate total dose effects for typical space applications. Appendix B contains the schematics and Verilog files of this design.

Table 2 lists measured electrical parameters and the corresponding logic design. The functionality is measured on the output pin (O\_BS) of a combinational buffer-string with 14,000 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 10,728 bits total, and half of the output pins (OUTX0, OUTX1, OUTX2, OUTX3, OUTX4, OUTX5, OUTX6 and OUTX7) of the embedded RAM configured as 16K×16.

$I_{CC}$  is measured on the power supply of the logic-array ( $I_{CCA}$ ) and I/O ( $I_{CCI}$ ) respectively. The input logic threshold ( $V_{IL}/V_{IH}$ ) is measured on single-ended inputs EN8, DA, IO\_I1, IO\_I2, IO\_I3, IO\_I4, IO\_I5 and IO\_I6, and also on differential inputs DIO\_I1P, DIO\_I2P, DIO\_I3P, DIO\_I4P, DIO\_I5P, DIO\_I6P and DIO\_I7P. The differential inputs are configured as LVPECL instead of LVDS because LVPECL, using 3.3 VDC, is worse than LVDS, which uses 2.5 VDC. 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 ( $V_{OL}/V_{OH}$ ) is measured on QA0. 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 captures.

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. $I_{CC}$ ( $I_{CCA}/I_{CCI}$ )	DUT power supply
3. Input Threshold ( $V_{IL}/V_{IH}$ )	Single ended inputs (EN8/YQ0, DA/QA0, IO_I1/IO_O1, IO_I2/IO_O2, IO_I3/IO_O3, IO_I4/IO_O4, IO_I5/IO_O5, IO_I6/IO_O6), and differential inputs (DIO_I1P/DIO_O1, DIO_I2P/DIO_O2, DIO_I3P/DIO_O3, DIO_I4P/DIO_O4, DIO_I5P/DIO_O5, DIO_I6P/DIO_O6, DIO_I7P/DIO_O7)
4. Output Drive ( $V_{OL}/V_{OH}$ )	Output buffer (DA/QA0)
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 ( $I_{CCA}$ and $I_{CCI}$ )

Figures 2-6 illustrate the plot of in-flux standby  $I_{CCA}$  and  $I_{CCI}$  versus total dose for every DUT. Table 3 summarizes the pre-irradiation, post-irradiation and post-annealing  $I_{CC}$ . The post-annealing  $I_{CC}$  for four different bit patterns, all '0', all '1', checkerboard and inverted-checkerboard, in the RAM are basically the same.

Table 3 Pre-irradiation, Post Irradiation and Post-Annealing  $I_{CC}$

DUT	Total Dose krad(SiO <sub>2</sub> )	$I_{CCA}$ (mA)			$I_{CCI}$ (mA)		
		Pre-irrad	Post-irrad	Post-ann	Pre-irrad	Post-irrad	Post-ann
7734	300	2	39	4	21	141	46
7741	300	12	64	16	21	154	45
7745	300	11	46	16	21	156	90
7756	200	9	11	9	22	68	32
7776	200	6	7	6	22	64	31

In compliance with TM1019.6 subsection 3.11.2.c, the post-irradiation-parametric limit (PIPL) for the post-annealing  $I_{CCI}$  in this test is defined as the addition of highest  $I_{CCI}$ ,  $I_{CCDA}$  and  $I_{CCDIFFA}$  values in Table 2-4 of the RTAXS spec sheet:

[http://www.actel.com/documents/RTAXS\\_DS.pdf](http://www.actel.com/documents/RTAXS_DS.pdf)

For  $I_{CCA}$ , the PIPL is 500 mA; the PIPL of  $I_{CCI}$  equals to  $35+10+3.13 \times 7 = 66.91$ (mA). Note that there are 7 pairs of differential LVPECL inputs in each DUT. Based on these PIPL, every DUT in every case passes both  $I_{CCA}$  and  $I_{CCI}$  specs.

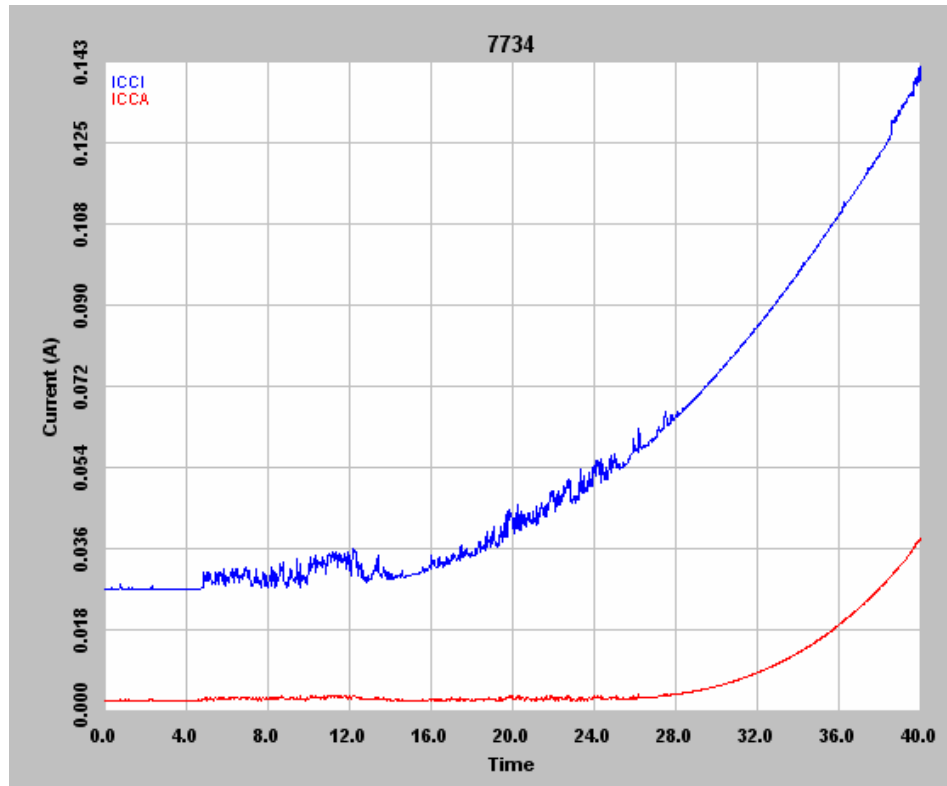


Figure 2 DUT 7734 in-flux I<sub>CCA</sub> and I<sub>CCI</sub>, time is measured in minutes.

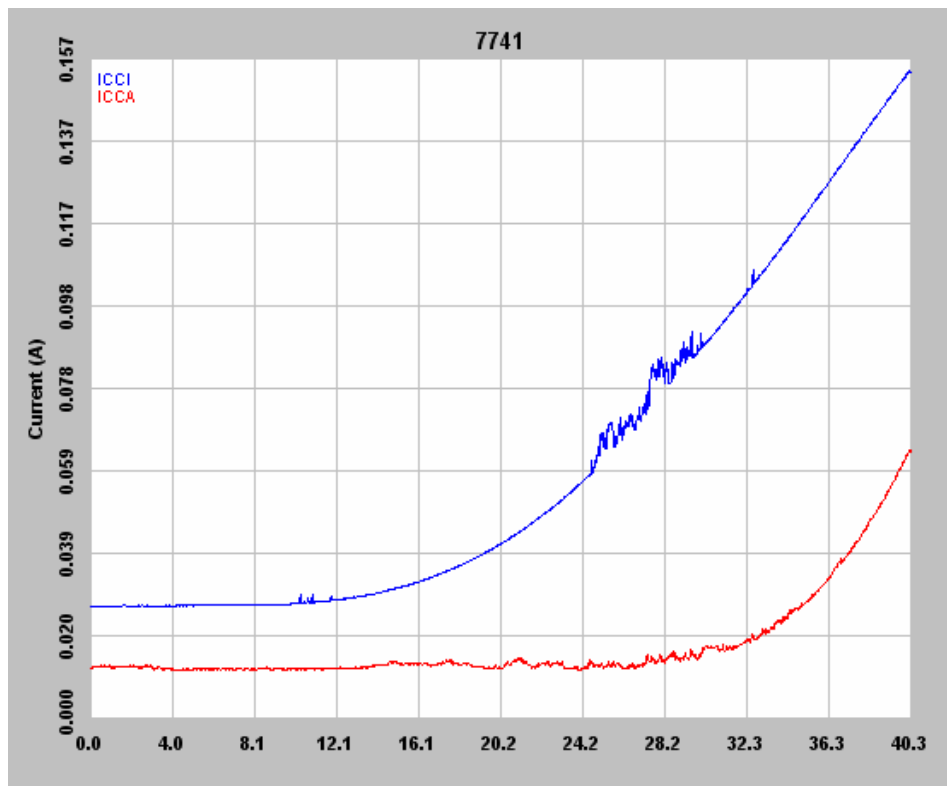


Figure 3 DUT 7741 in-flux I<sub>CCA</sub> and I<sub>CCI</sub>, time is measured in minutes.

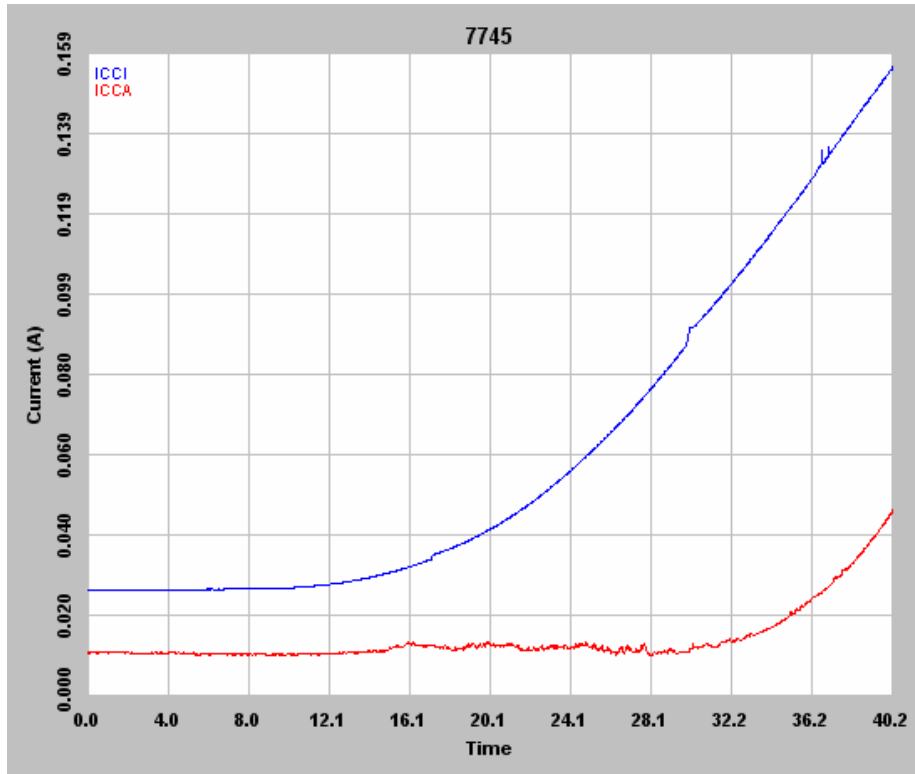


Figure 4 DUT 7745 in-flux  $I_{CCA}$  and  $I_{CCI}$ , time is measured in minutes.

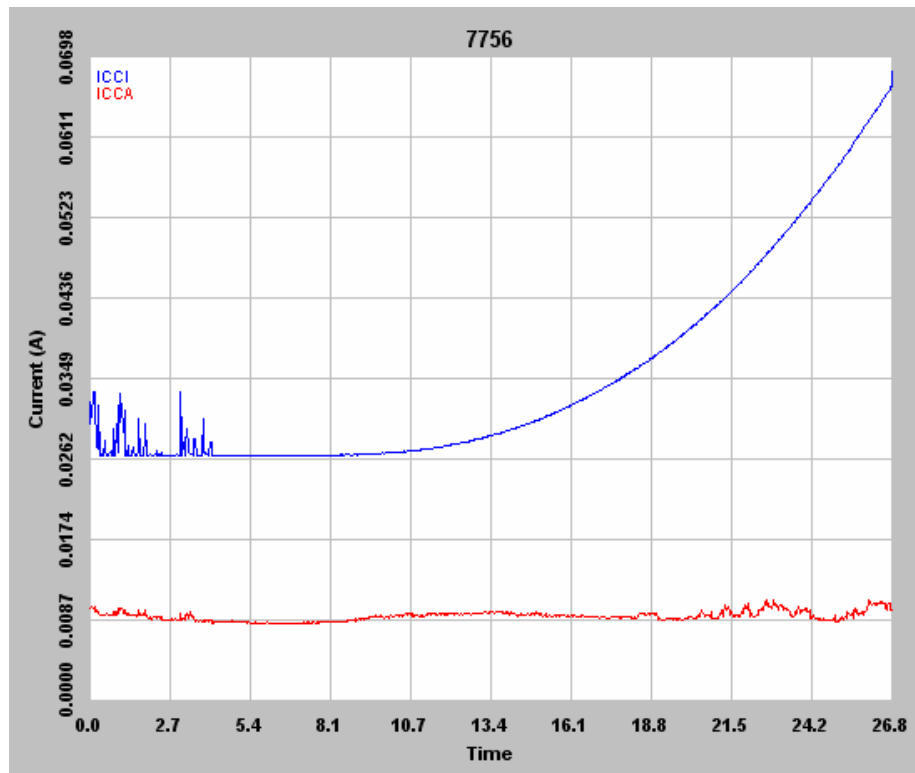


Figure 5 DUT 7756 in-flux  $I_{CCA}$  and  $I_{CCI}$ , time is measured in minutes.

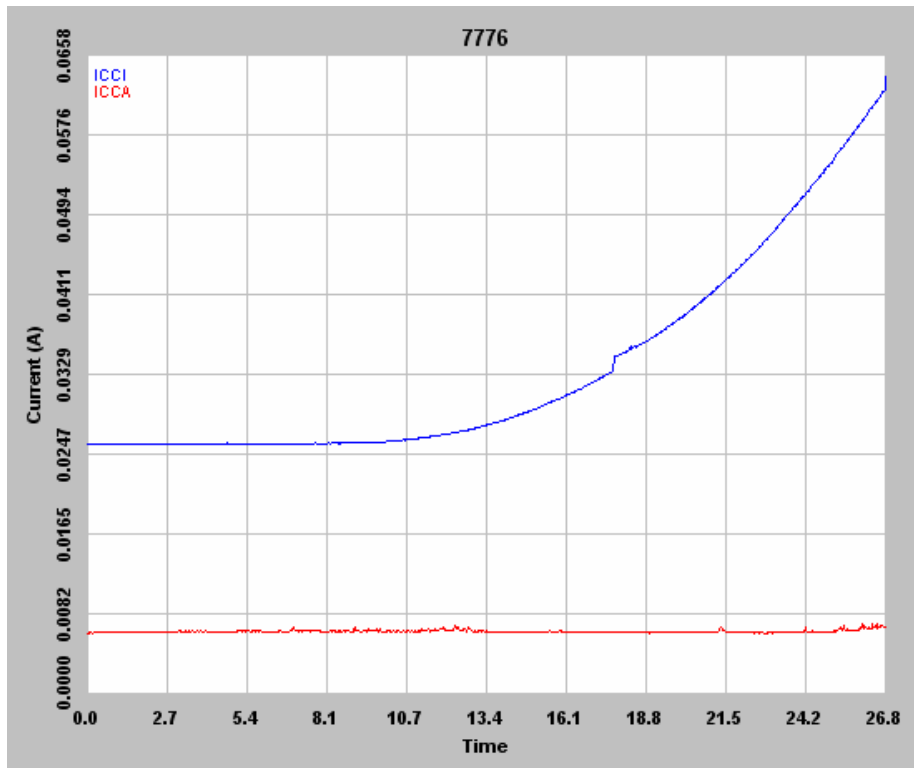


Figure 6 DUT 7776 in-flux  $I_{CCA}$  and  $I_{CCI}$ , time is measured in minutes.

C. Single-Ended Input Logic Threshold ( $V_{IL}/V_{IH}$ )

Table 4 lists the pre-irradiation and post-annealing single-ended input logic threshold. All data are within the spec 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_I1/IO_O1				IO_I2/IO_O2			
DUT	Total Dose	Pre-Irrad	Post-Ann	Pre-Irrad	Post-Ann	Pre-Irrad	Post-Ann	Pre-Irrad	Post-Ann
		$V_{IL}$ (V)		$V_{IH}$ (V)		$V_{IL}$ (V)		$V_{IH}$ (V)	
7734	300 krad	1.51	1.51	1.49	1.49	1.52	1.50	1.50	1.49
7741	300 krad	1.52	1.51	1.49	1.52	1.52	1.50	1.53	1.48
7745	300 krad	1.52	1.52	1.53	1.49	1.51	1.51	1.53	1.53
7756	200 krad	1.52	1.51	1.49	1.49	1.52	1.50	1.53	1.53
7776	200 krad	1.52	1.52	1.49	1.49	1.51	1.50	1.53	1.48

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

In/Out Pin:		IO_I3/IO_O3				IO_I4/IO_O4			
DUT	Total Dose	Pre-Irrad	Post-Ann	Pre-Irrad	Post-Ann	Pre-Irrad	Post-Ann	Pre-Irrad	Post-Ann
		$V_{IL}$ (V)		$V_{IH}$ (V)		$V_{IL}$ (V)		$V_{IH}$ (V)	
7734	300 krad	1.51	1.48	1.48	1.51	1.50	1.50	1.49	1.48
7741	300 krad	1.48	1.47	1.50	1.49	1.51	1.50	1.52	1.52
7745	300 krad	1.48	1.47	1.51	1.50	1.51	1.51	1.49	1.52
7756	200 krad	1.48	1.48	1.51	1.50	1.51	1.51	1.51	1.50
7776	200 krad	1.48	1.47	1.51	1.50	1.51	1.51	1.52	1.51

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

In/Out Pin:		IO_I5/IO_O5				IO_I6/IO_O6			
DUT	Total Dose	Pre-Irrad	Post-Ann	Pre-Irrad	Post-Ann	Pre-Irrad	Post-Ann	Pre-Irrad	Post-Ann
		$V_{IL}$ (V)		$V_{IH}$ (V)		$V_{IL}$ (V)		$V_{IH}$ (V)	
7734	300 krad	1.50	1.53	1.48	1.50	1.52	1.51	1.48	1.48
7741	300 krad	1.52	1.51	1.48	1.50	1.52	1.51	1.50	1.49
7745	300 krad	1.53	1.51	1.48	1.50	1.52	1.52	1.50	1.49
7756	200 krad	1.53	1.50	1.48	1.50	1.52	1.51	1.50	1.49
7776	200 krad	1.53	1.52	1.50	1.50	1.52	1.51	1.50	1.48

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
		$V_{IL}$ (V)		$V_{IH}$ (V)		$V_{IL}$ (V)		$V_{IH}$ (V)	
7734	300 krad	1.51	1.51	1.49	1.48	1.53	1.50	1.48	1.49
7741	300 krad	1.52	1.51	1.50	1.49	1.50	1.53	1.49	1.47
7745	300 krad	1.52	1.52	1.50	1.49	1.51	1.53	1.49	1.48
7756	200 krad	1.52	1.51	1.50	1.49	1.51	1.50	1.50	1.48
7776	200 krad	1.52	1.52	1.50	1.49	1.52	1.50	1.48	1.49



D. Differential Input (LVPECL) Threshold Voltage ( $V_{IL}/V_{IH}$ )

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

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

In/Out Pin:		DIO_I1P/DIO_O1				DIO_I2P/DIO_O2			
DUT	Total Dose	Pre-Irrad	Post-Ann	Pre-Irrad	Post-Ann	Pre-Irrad	Post-Ann	Pre-Irrad	Post-Ann
		$V_{IL}$ (V)		$V_{IH}$ (V)		$V_{IL}$ (V)		$V_{IH}$ (V)	
7734	300 krad	1.78	1.78	1.78	1.78	1.80	1.79	1.78	1.78
7741	300 krad	1.79	1.79	1.78	1.78	1.79	1.79	1.78	1.78
7745	300 krad	1.79	1.79	1.77	1.77	1.79	1.79	1.78	1.78
7756	200 krad	1.79	1.79	1.78	1.78	1.79	1.79	1.78	1.78
7776	200 krad	1.77	1.77	1.78	1.77	1.79	1.78	1.77	1.77

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

In/Out Pin:		DIO_I3P/DIO_O3				DIO_I4P/DIO_O4			
DUT	Total Dose	Pre-Irrad	Post-Ann	Pre-Irrad	Post-Ann	Pre-Irrad	Post-Ann	Pre-Irrad	Post-Ann
		$V_{IL}$ (V)		$V_{IH}$ (V)		$V_{IL}$ (V)		$V_{IH}$ (V)	
7734	300 krad	1.79	1.79	1.79	1.78	1.79	1.77	1.77	1.77
7741	300 krad	1.79	1.80	1.78	1.78	1.77	1.77	1.78	1.79
7745	300 krad	1.79	1.79	1.78	1.78	1.78	1.77	1.78	1.79
7756	200 krad	1.78	1.79	1.77	1.77	1.77	1.80	1.78	1.79
7776	200 krad	1.79	1.79	1.78	1.77	1.80	1.77	1.77	1.78

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

In/Out Pin:		DIO_I5P/DIO_O5				DIO_I6P/DIO_O6			
DUT	Total Dose	Pre-Irrad	Post-Ann	Pre-Irrad	Post-Ann	Pre-Irrad	Post-Ann	Pre-Irrad	Post-Ann
		$V_{IL}$ (V)		$V_{IH}$ (V)		$V_{IL}$ (V)		$V_{IH}$ (V)	
7734	300 krad	1.79	1.79	1.78	1.78	1.78	1.78	1.79	1.80
7741	300 krad	1.80	1.79	1.78	1.78	1.79	1.79	1.80	1.80
7745	300 krad	1.79	1.79	1.78	1.78	1.82	1.79	1.78	1.78
7756	200 krad	1.79	1.79	1.78	1.78	1.78	1.78	1.80	1.80
7776	200 krad	1.79	1.79	1.78	1.78	1.78	1.78	1.80	1.80

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

In/Out Pin:		DIO_I7P/DIO_O7			
DUT	Total Dose	Pre-Irrad	Post-Ann	Pre-Irrad	Post-Ann
		$V_{IL}$ (V)		$V_{IH}$ (V)	
7734	300 krad	1.80	1.80	1.79	1.80
7741	300 krad	1.80	1.80	1.79	1.79
7745	300 krad	1.79	1.79	1.78	1.79
7756	200 krad	1.79	1.79	1.79	1.78
7776	200 krad	1.79	1.79	1.78	1.78

### E. Output-Drive Voltage ( $V_{OL}/V_{OH}$ )

The pre-irradiation and post-annealing  $V_{OL}/V_{OH}$  are listed in Tables 6 and 7. The post-annealing data are within the spec limits; in each case, the radiation-induced degradation is well within 10%.

Table 6 Pre-Irradiation and Post-Annealing  $V_{OL}$  (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
7734	300 krad	0.006	0.005	0.079	0.075	0.132	0.126	0.336	0.321	0.708	0.675
7741	300 krad	0.005	0.005	0.078	0.074	0.131	0.125	0.332	0.318	0.699	0.669
7745	300 krad	0.005	0.005	0.079	0.075	0.132	0.126	0.335	0.321	0.705	0.675
7756	200 krad	0.006	0.005	0.079	0.076	0.133	0.128	0.339	0.325	0.713	0.684
7776	200 krad	0.006	0.006	0.085	0.081	0.142	0.137	0.361	0.347	0.758	0.729

Table 7 Pre-Irradiation and Post-Annealing  $V_{OH}$  (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
7734	300 krad	3.29	3.28	3.22	3.22	3.12	3.11	2.83	2.83	2.27	2.26
7741	300 krad	3.29	3.28	3.23	3.22	3.12	3.12	2.84	2.84	2.29	2.28
7745	300 krad	3.29	3.28	3.22	3.22	3.12	3.11	2.84	2.83	2.28	2.27
7756	200 krad	3.29	3.29	3.22	3.22	3.12	3.11	2.83	2.83	2.28	2.27
7776	200 krad	3.29	3.28	3.22	3.22	3.11	3.10	2.81	2.81	2.23	2.23

### F. Propagation Delay

Table 8 lists the pre-irradiation and post-annealing propagation delays, and percentage change (degradations) for each DUT. Every DUT passes the  $\pm 10\%$ -degradation criterion.

Table 8 Radiation-Induced Propagation Delay Degradations

DUT	Total Dose	Pre-Irradiation (us)	Post-Annealing (us)	Degradation (%)
7734	300 krad	6.771	6.711	-0.89%
7741	300 krad	6.241	6.185	-0.89%
7745	300 krad	6.287	6.242	-0.72%
7756	200 krad	6.330	6.264	-1.05%
7776	200 krad	6.367	6.296	-1.12%

### G. Transition Time

Figures 7 to 18 show the pre-irradiation and post-annealing transition edges. In each case, the radiation-induced transition-time degradation is insignificant.

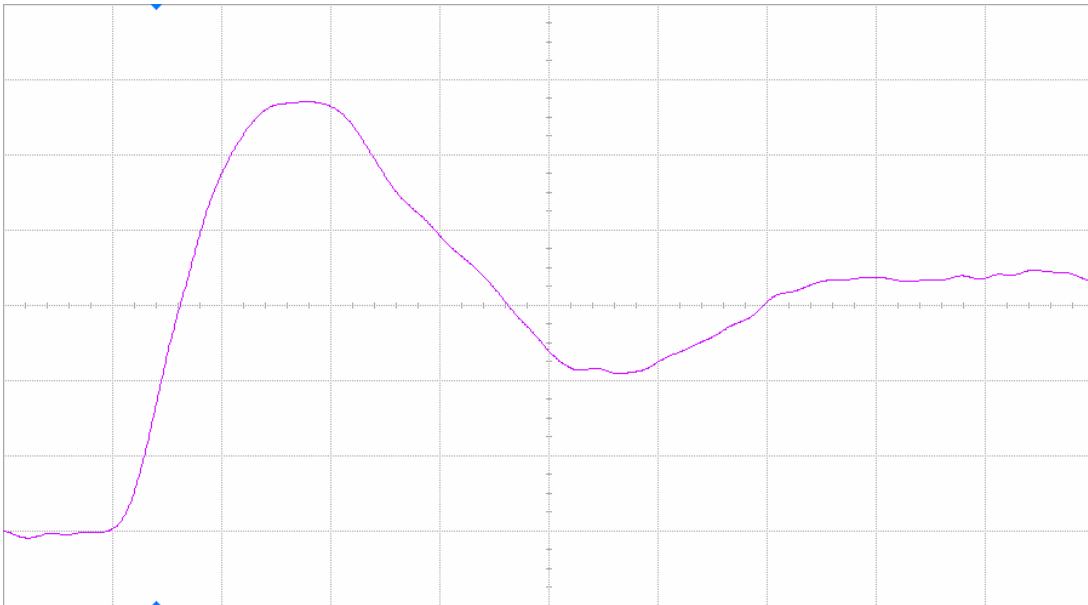


Figure 7(a) DUT 7734 pre-irradiation rising edge, abscissa scale is 1 V/div and ordinate scale is 2 ns/div.

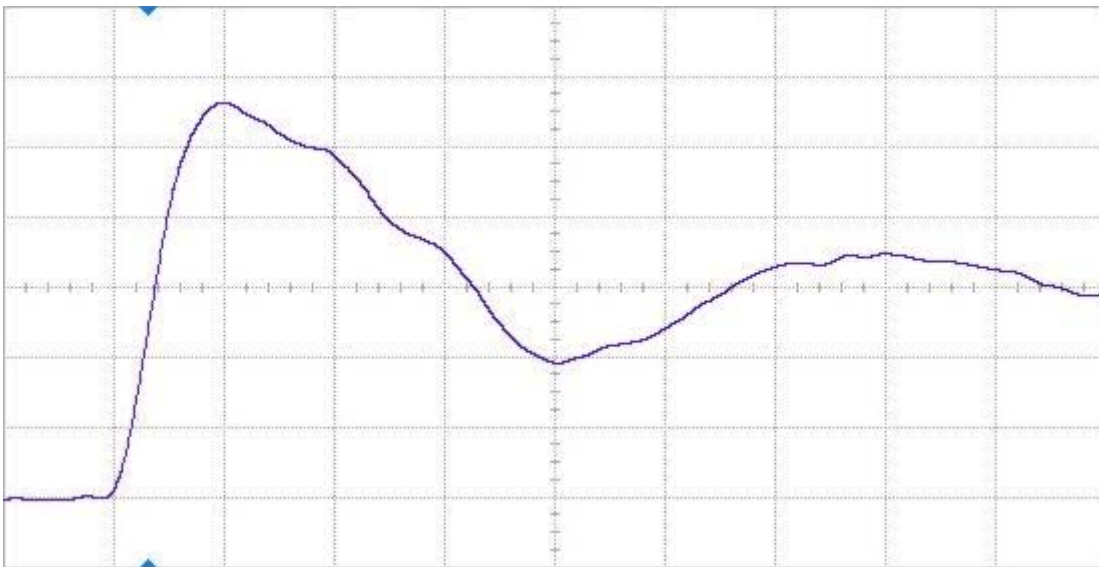


Figure 7(b) DUT 7734 post-annealing rising edge, abscissa scale is 1 V/div and ordinate scale is 2 ns/div.

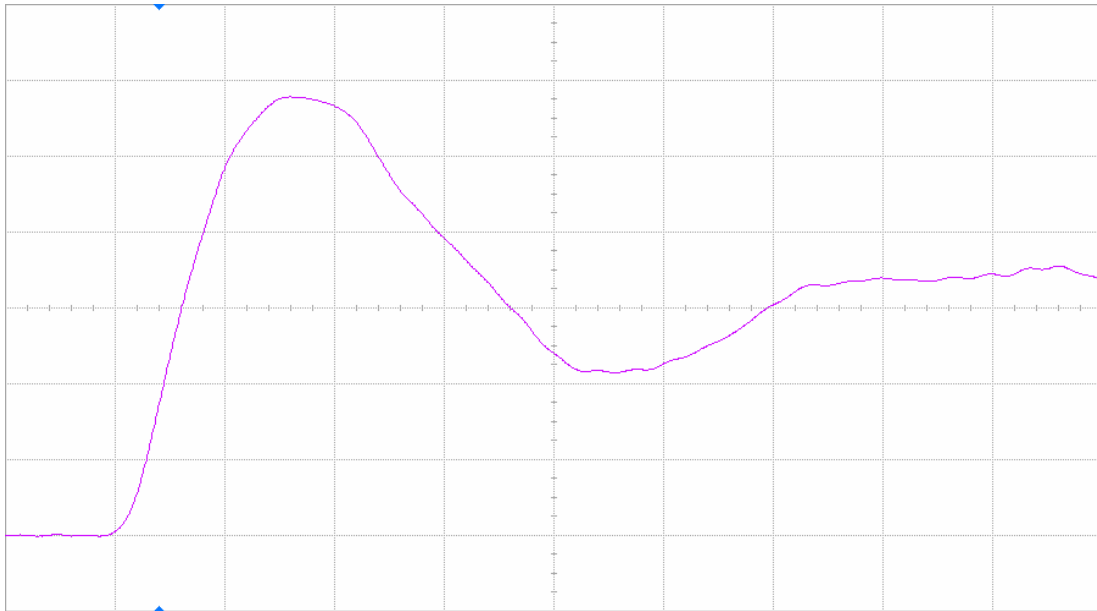


Figure 8(a) DUT 7741 pre-irradiation rising edge, abscissa scale is 1 V/div and ordinate scale is 2 ns/div.

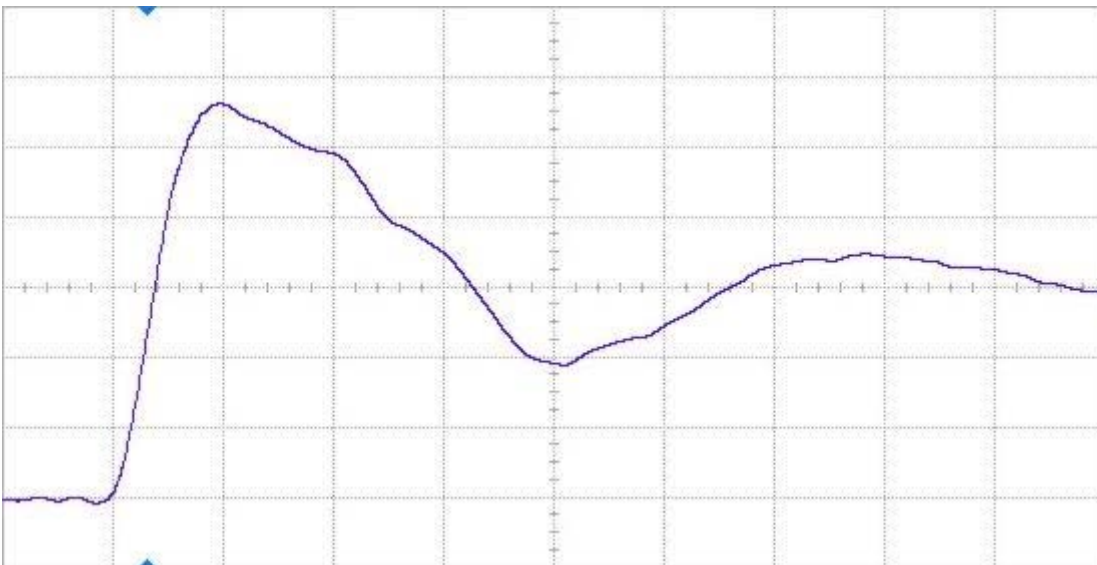


Figure 8(b) DUT 7741 post-annealing rising edge, abscissa scale is 1 V/div and ordinate scale is 2 ns/div.

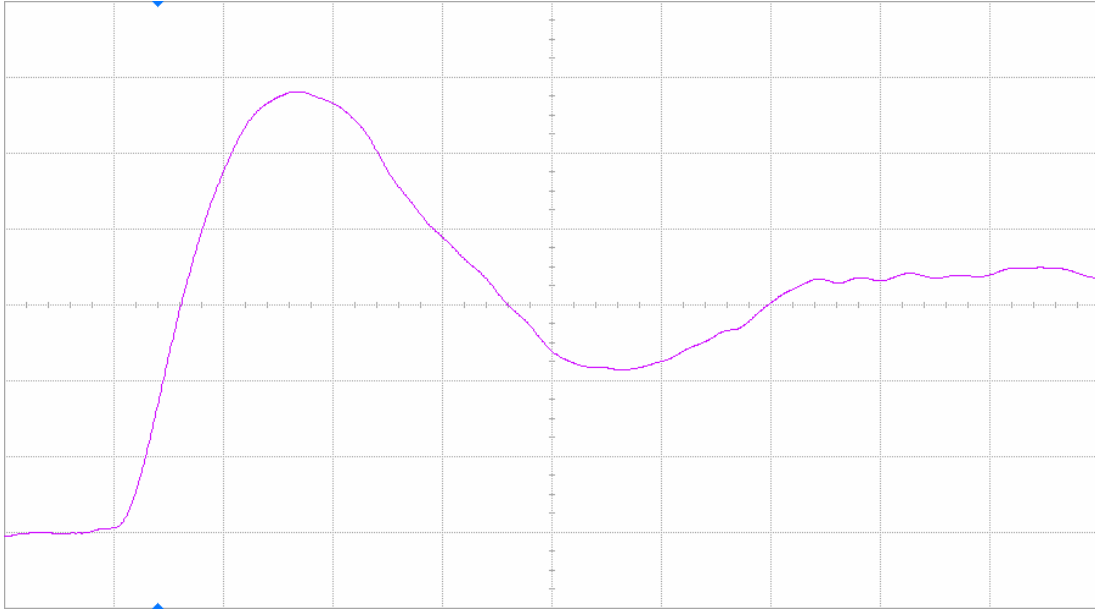


Figure 9(a) DUT 7745 pre-radiation rising edge, abscissa scale is 1 V/div and ordinate scale is 2 ns/div.

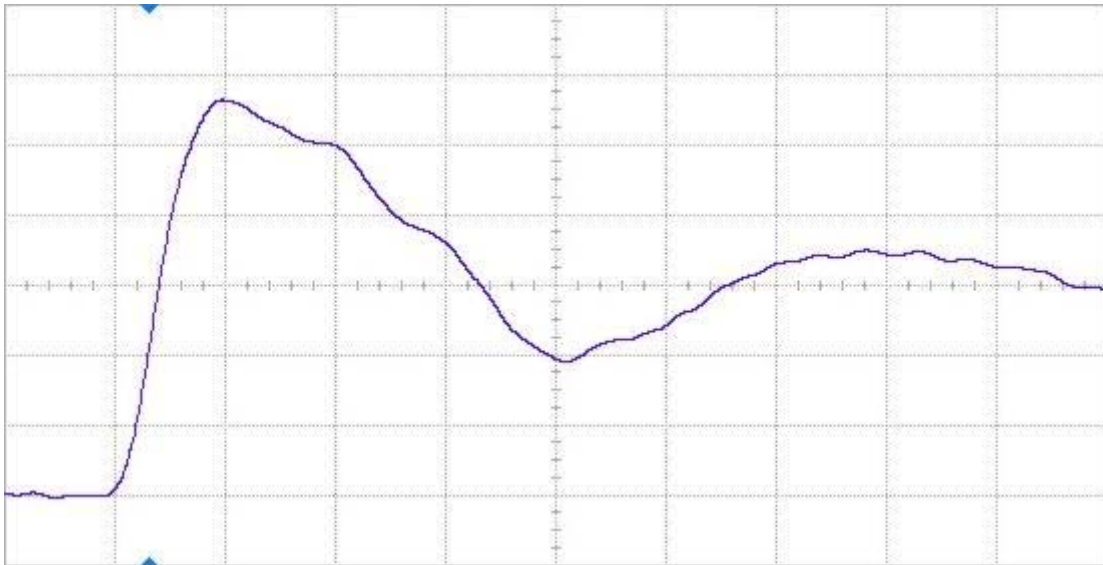


Figure 9(b) DUT 7745 post-annealing rising edge, abscissa scale is 1 V/div and ordinate scale is 2 ns/div.

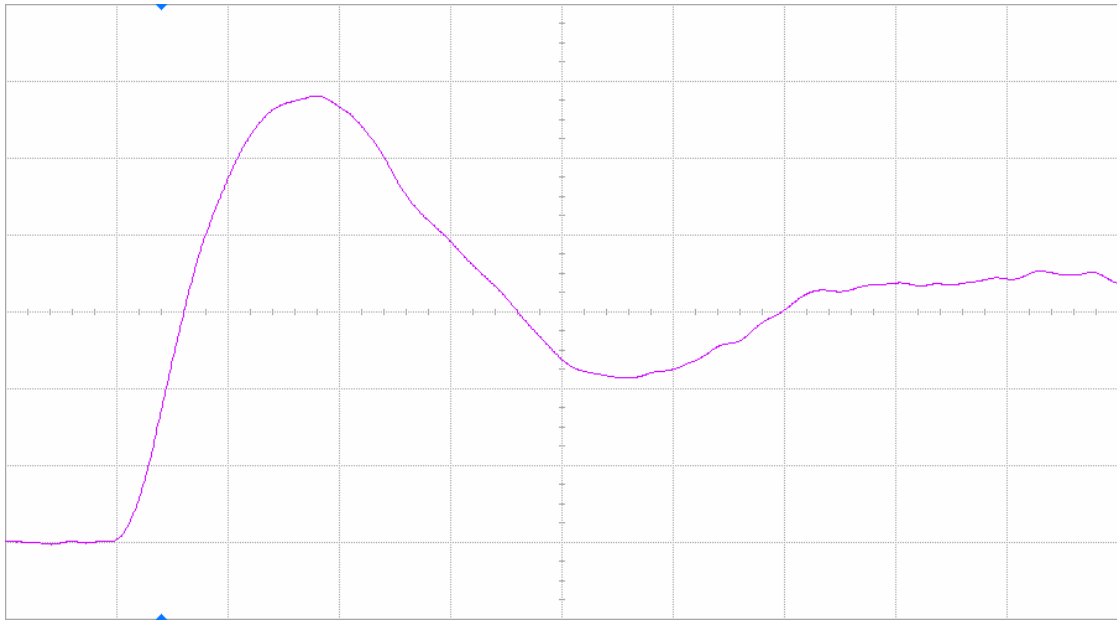


Figure 10(a) DUT 7756 pre-irradiation rising edge, abscissa scale is 1 V/div and ordinate scale is 2 ns/div.

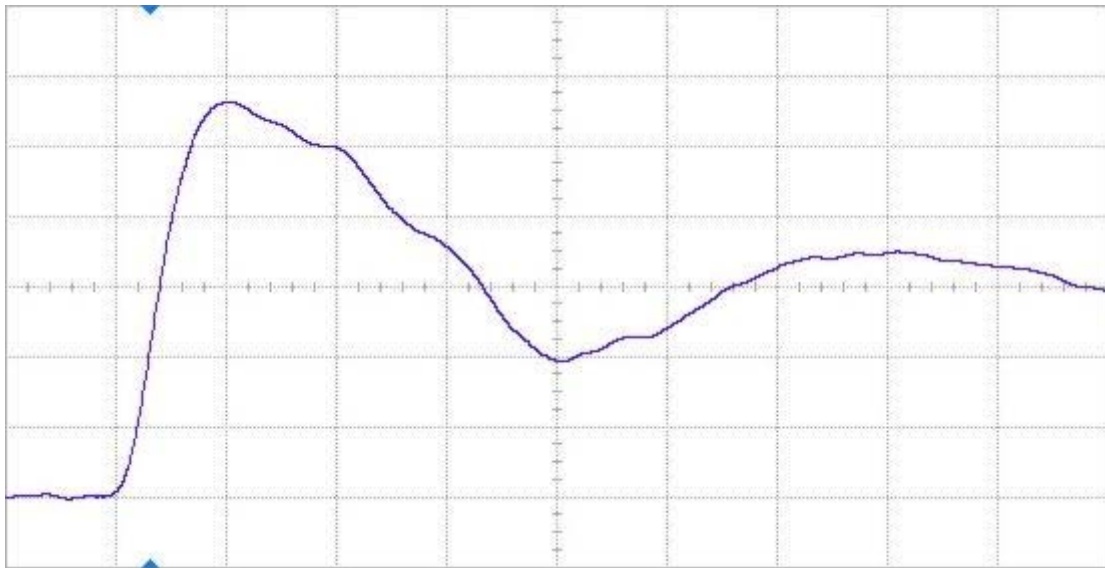


Figure 10(b) DUT 7756 post-annealing rising edge, abscissa scale is 1 V/div and ordinate scale is 2 ns/div.

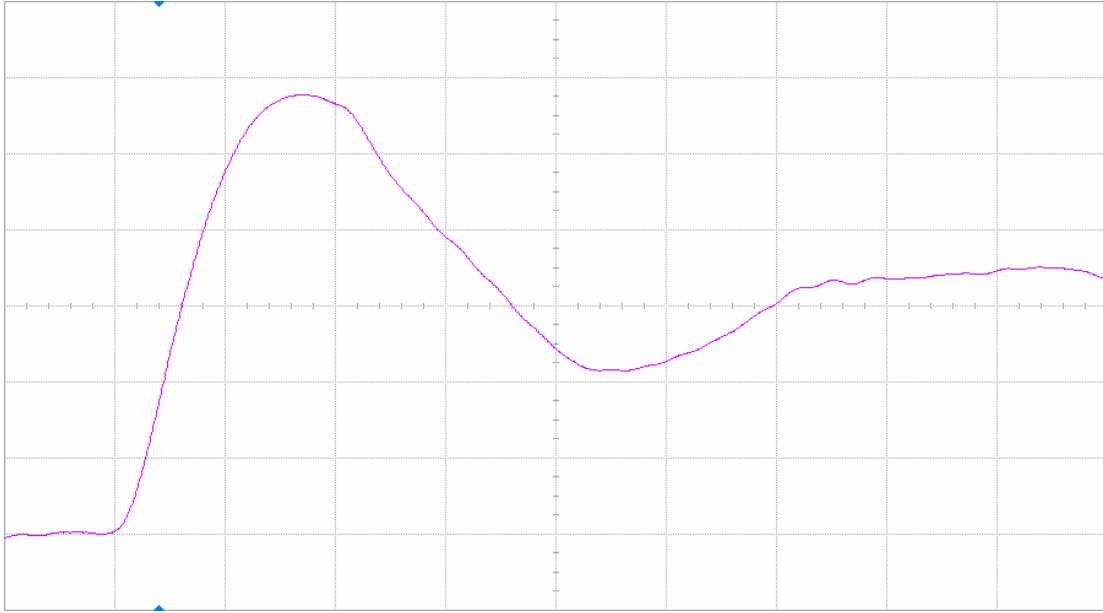


Figure 11(a) DUT 7776 pre-irradiation rising edge, abscissa scale is 1 V/div and ordinate scale is 2 ns/div.

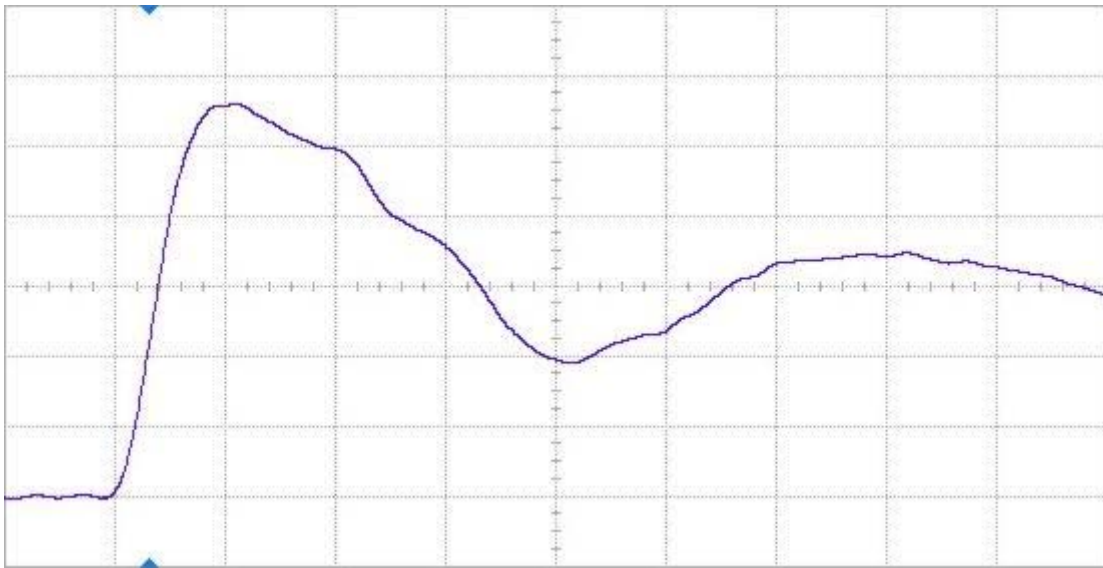


Figure 11(b) DUT 7776 post-annealing rising edge, abscissa scale is 1 V/div and ordinate scale is 2 ns/div.

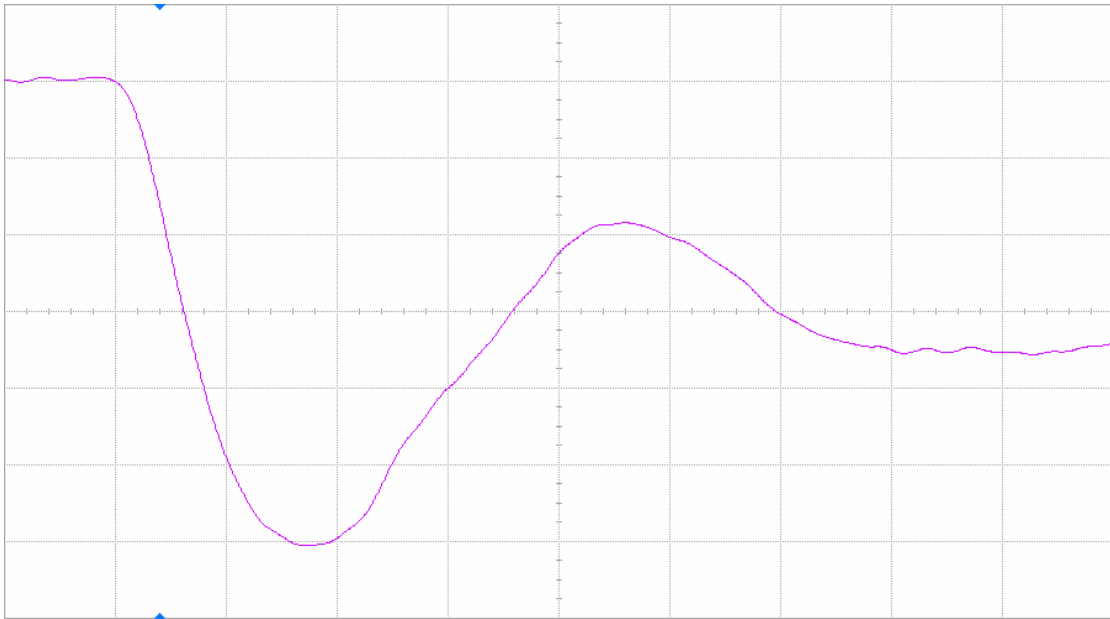


Figure 12(a) DUT 7734 pre-irradiation falling edge, abscissa scale is 1 V/div and ordinate scale is 2 ns/div.

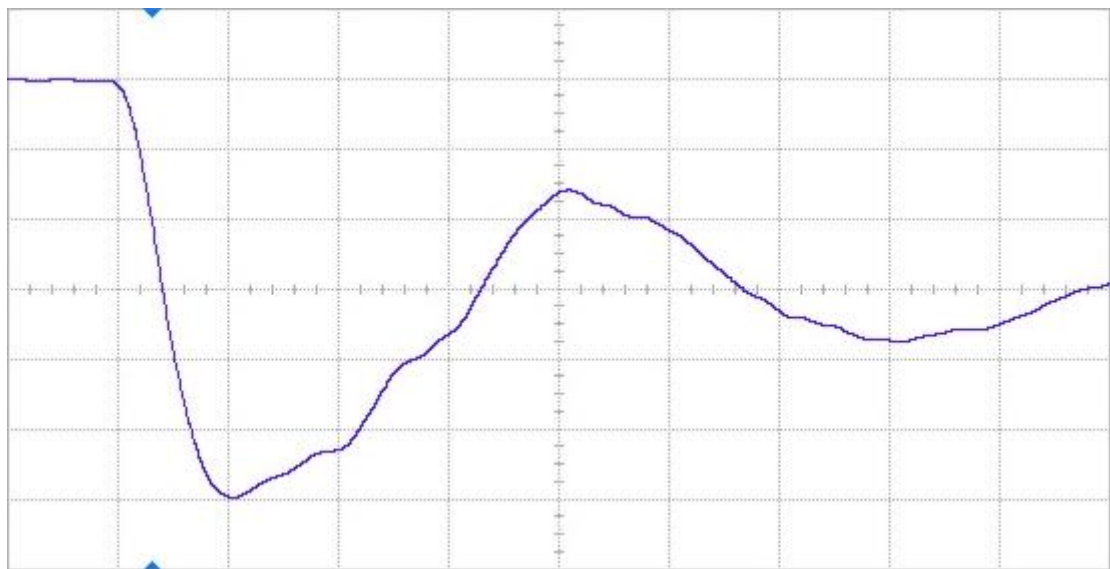


Figure 12(b) DUT 7734 post-annealing falling edge, abscissa scale is 1 V/div and ordinate scale is 2 ns/div.



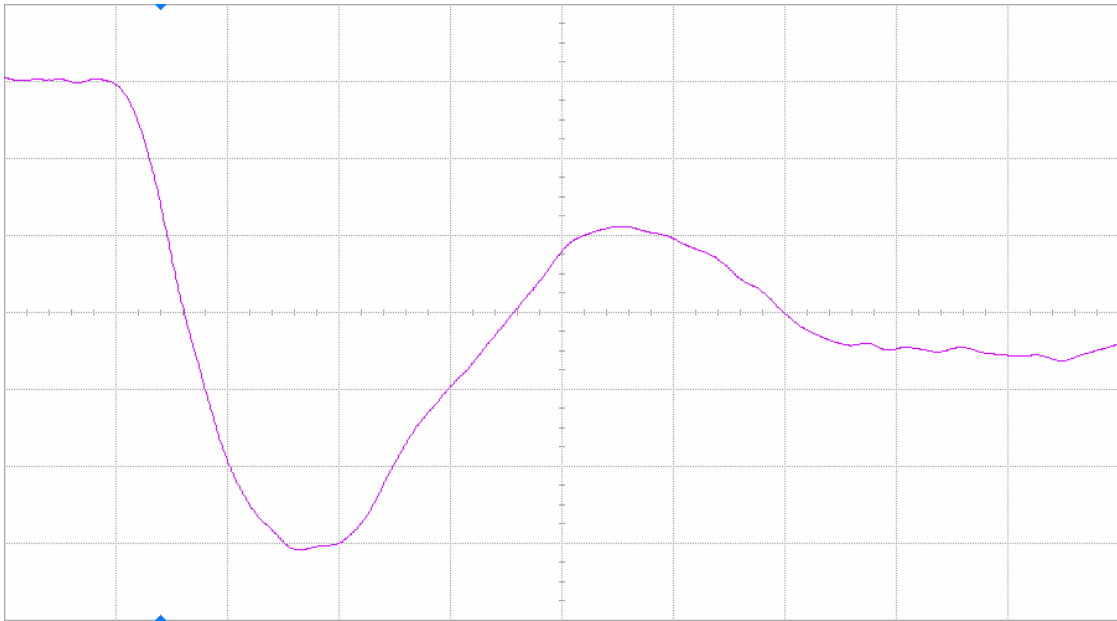


Figure 13(a) DUT 7741 pre-irradiation falling edge, abscissa scale is 1 V/div and ordinate scale is 2 ns/div.

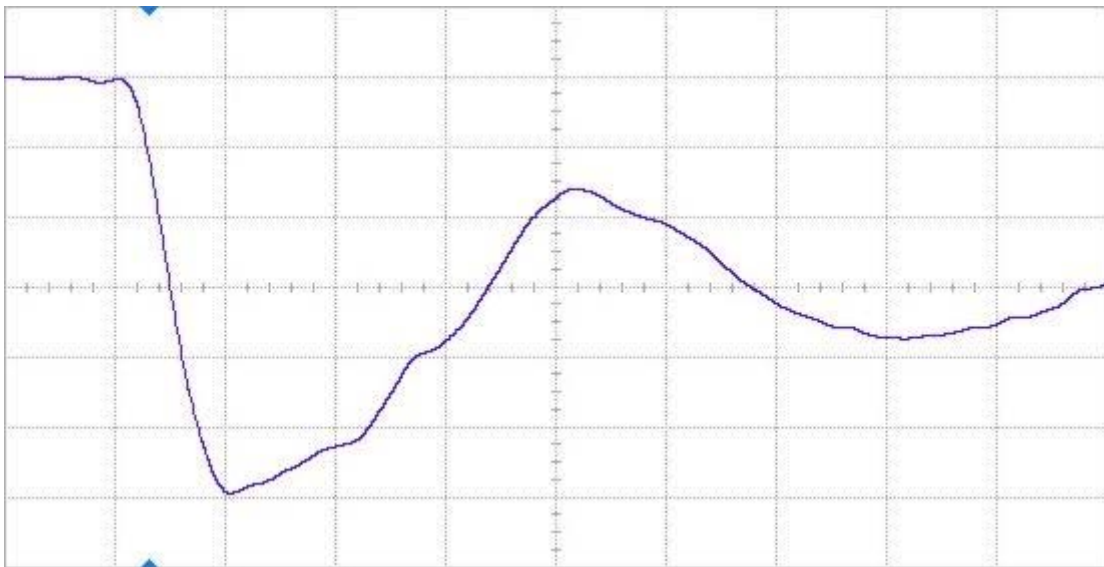


Figure 13(b) DUT 7741 post-annealing falling edge, abscissa scale is 1 V/div and ordinate scale is 2 ns/div.

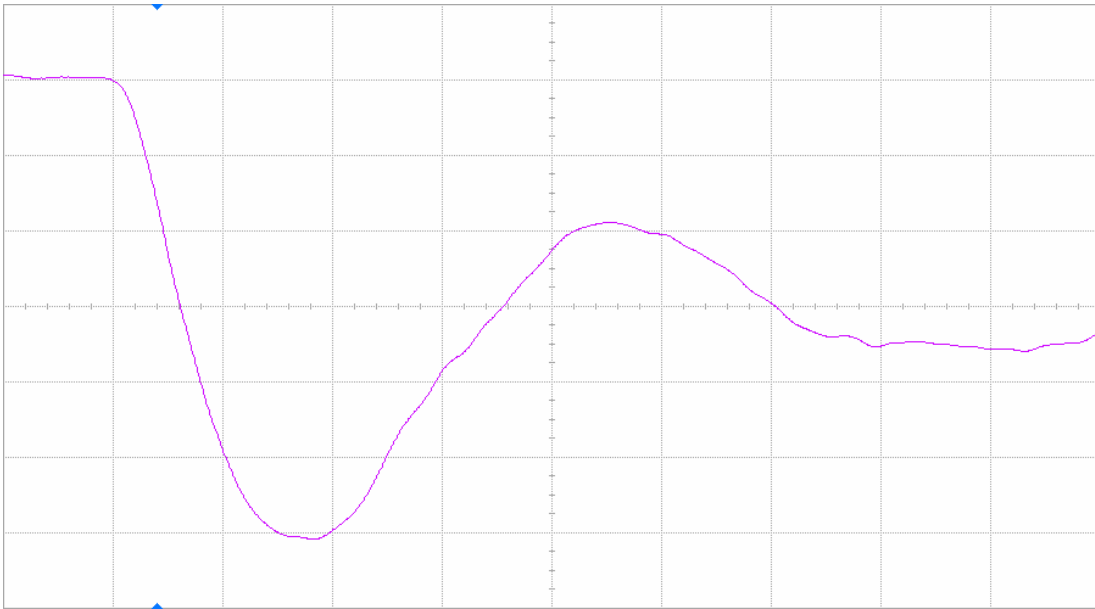


Figure 14(a) DUT 7745 pre-irradiation falling edge, abscissa scale is 1 V/div and ordinate scale is 2 ns/div.

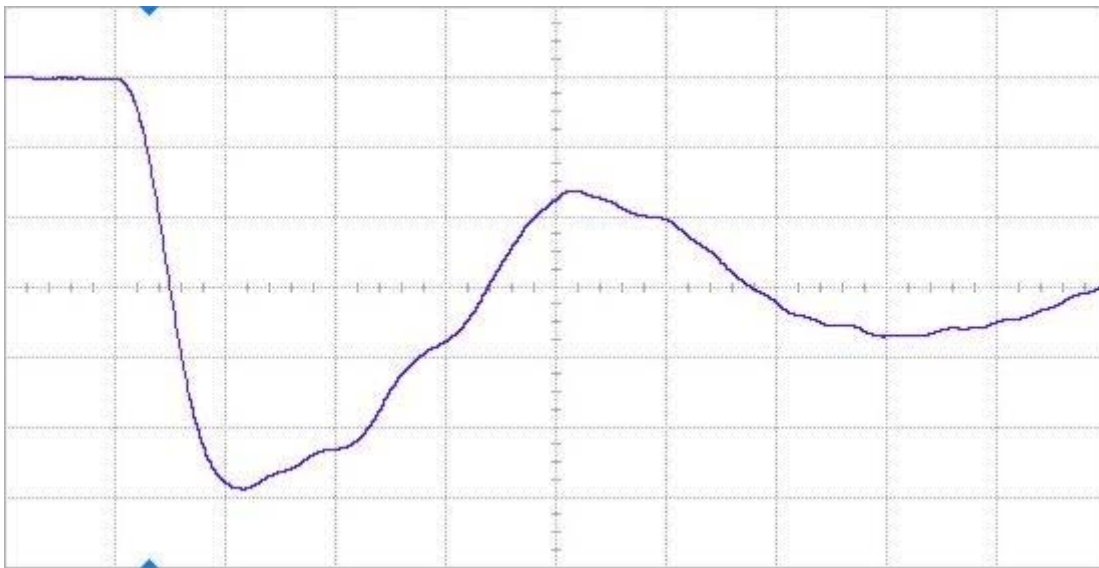


Figure 14(b) DUT 7745 post-annealing falling edge, abscissa scale is 1 V/div and ordinate scale is 2 ns/div.

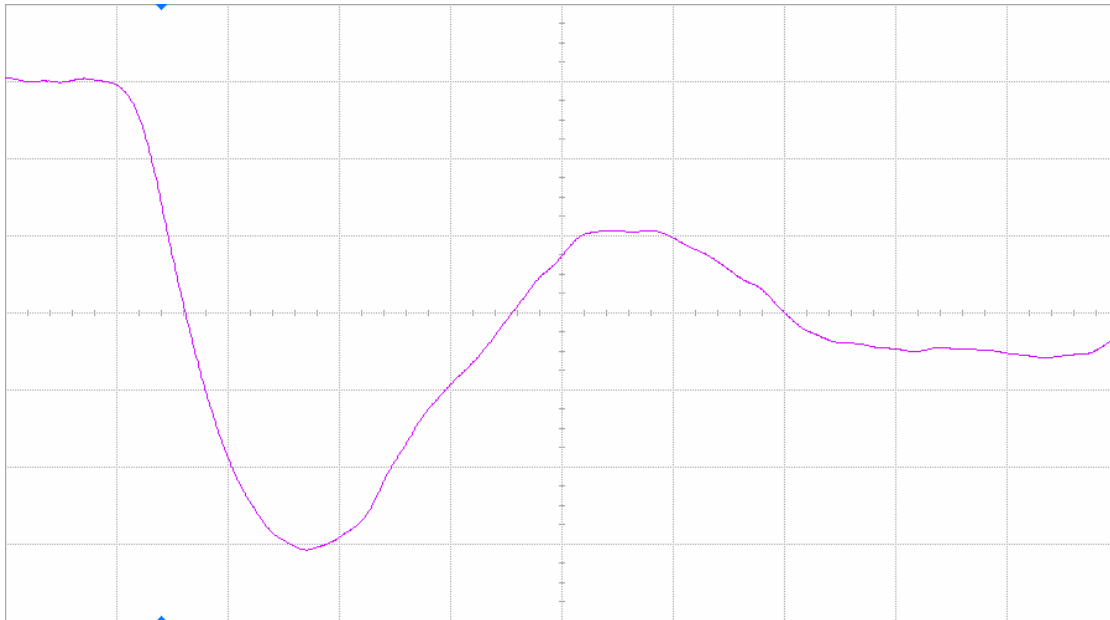


Figure 15(a) DUT 7756 pre-irradiation falling edge, abscissa scale is 1 V/div and ordinate scale is 2 ns/div.

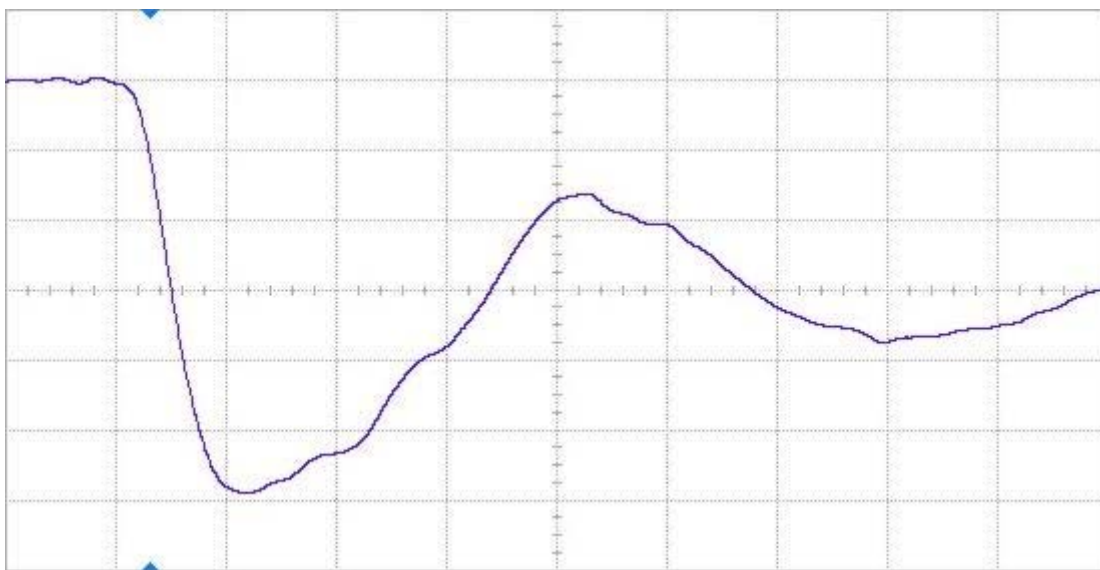


Figure 15(b) DUT 7756 post-annealing falling edge, abscissa scale is 1 V/div and ordinate scale is 2 ns/div.

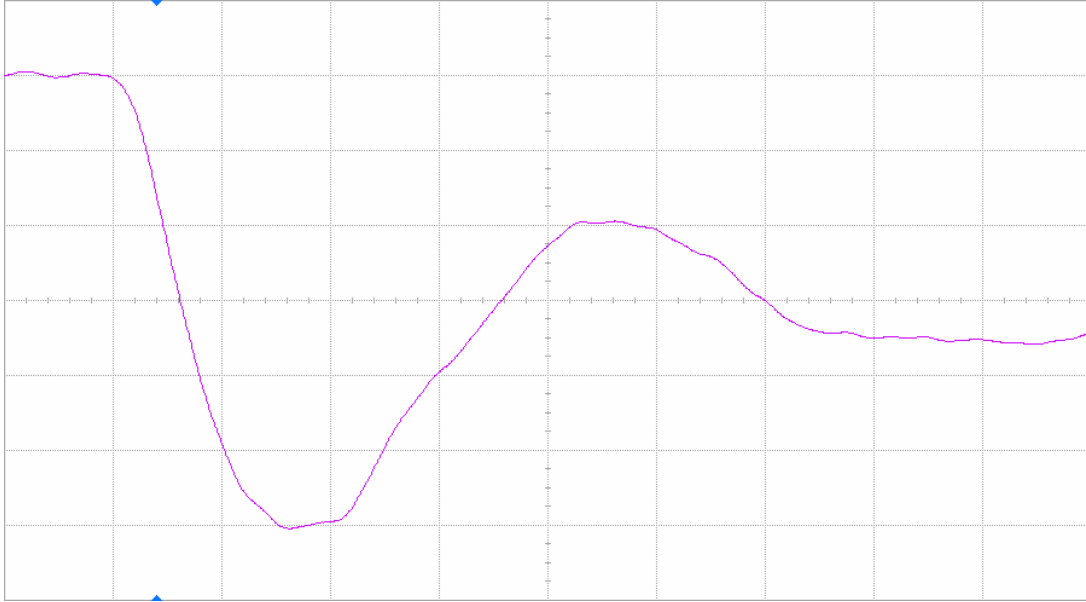


Figure 16(a) DUT 7776 pre-irradiation falling edge, abscissa scale is 1 V/div and ordinate scale is 2 ns/div.

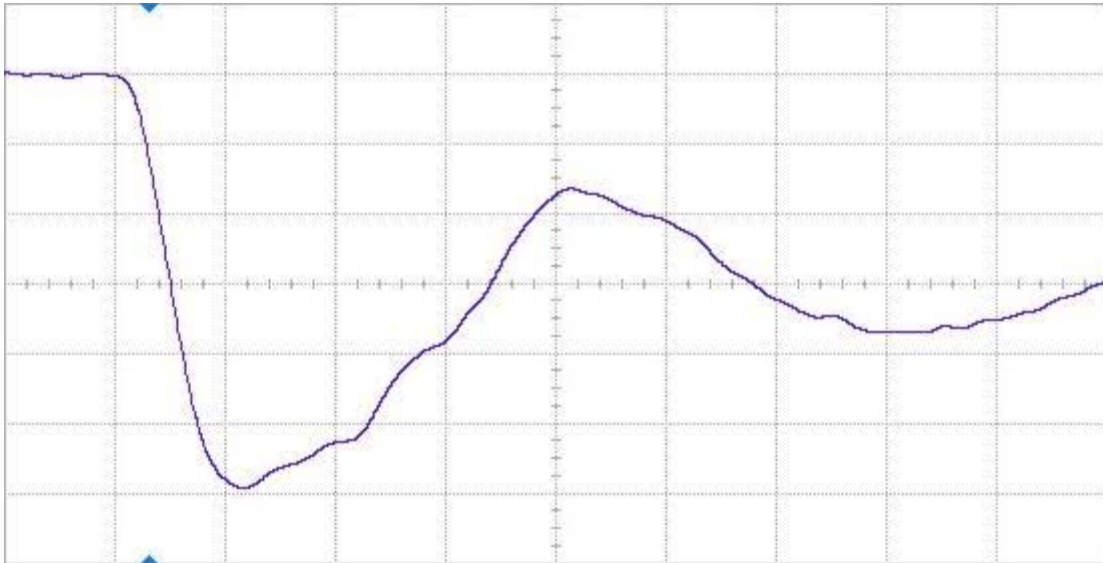


Figure 16(b) DUT 7776 post-annealing falling edge, abscissa scale is 1 V/div and ordinate scale is 2 ns/div.



Figure A1 IO bias during irradiation

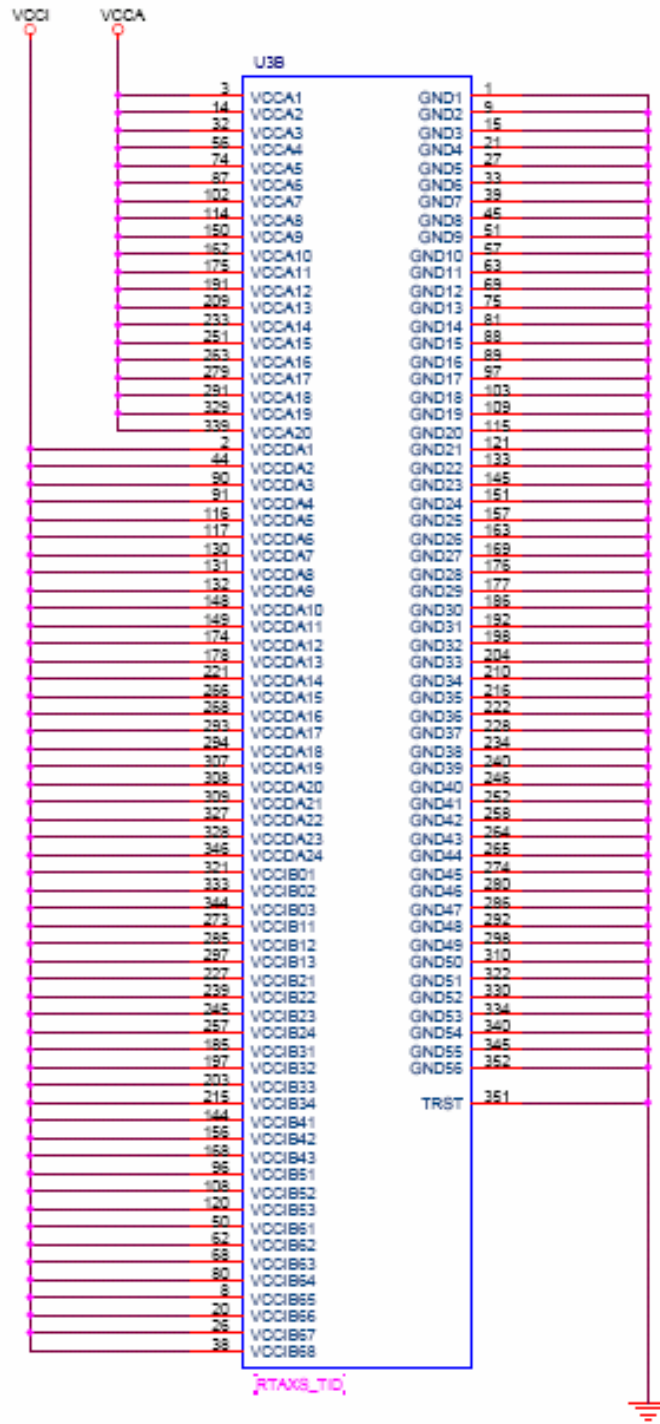
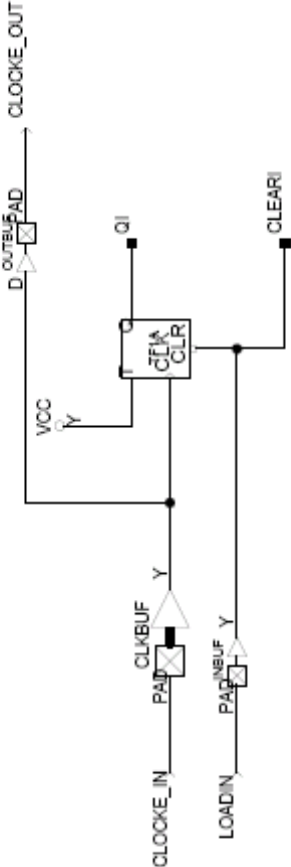
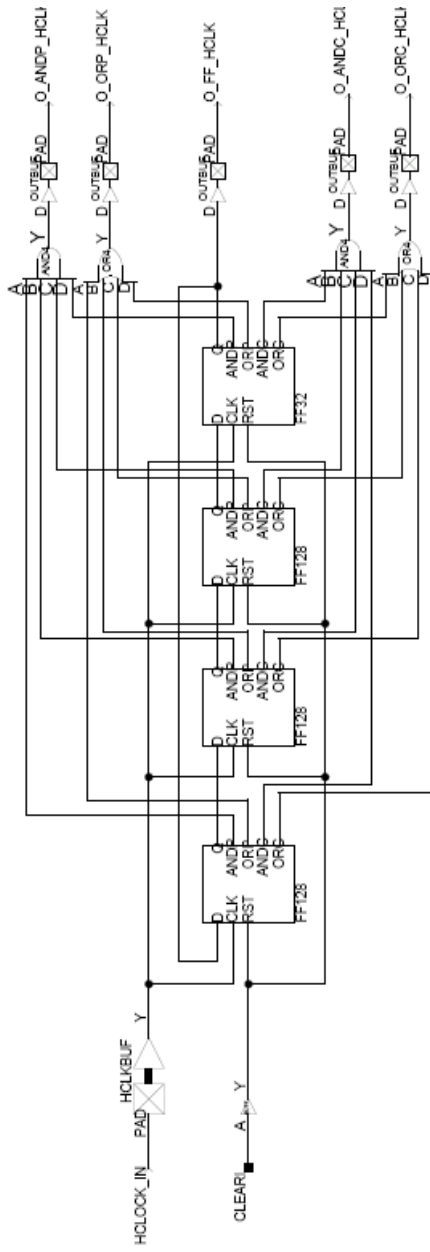


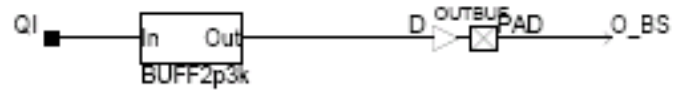
Figure A2 Power supply, ground and special pins bias during irradiation

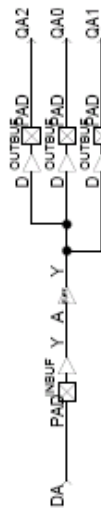
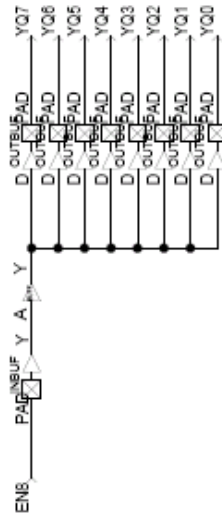
APPENDIX B DUT DESIGN SCHEMATICS AND VERILOG FILES

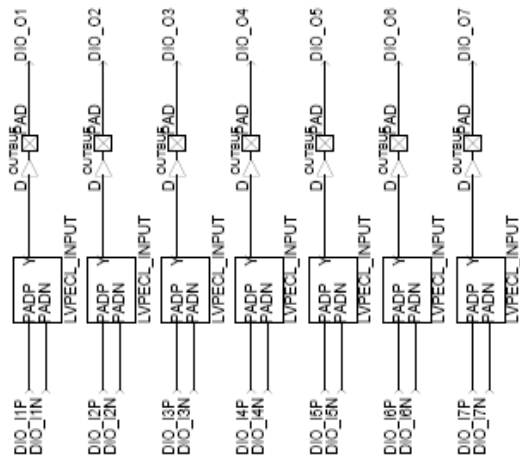
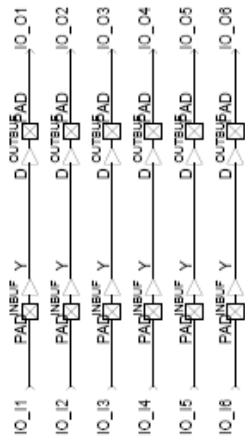


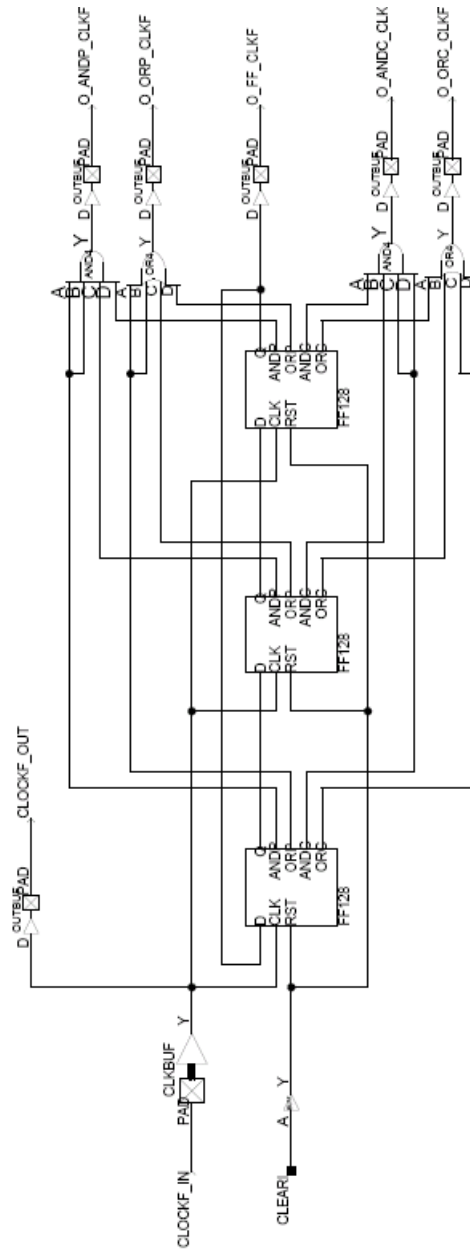


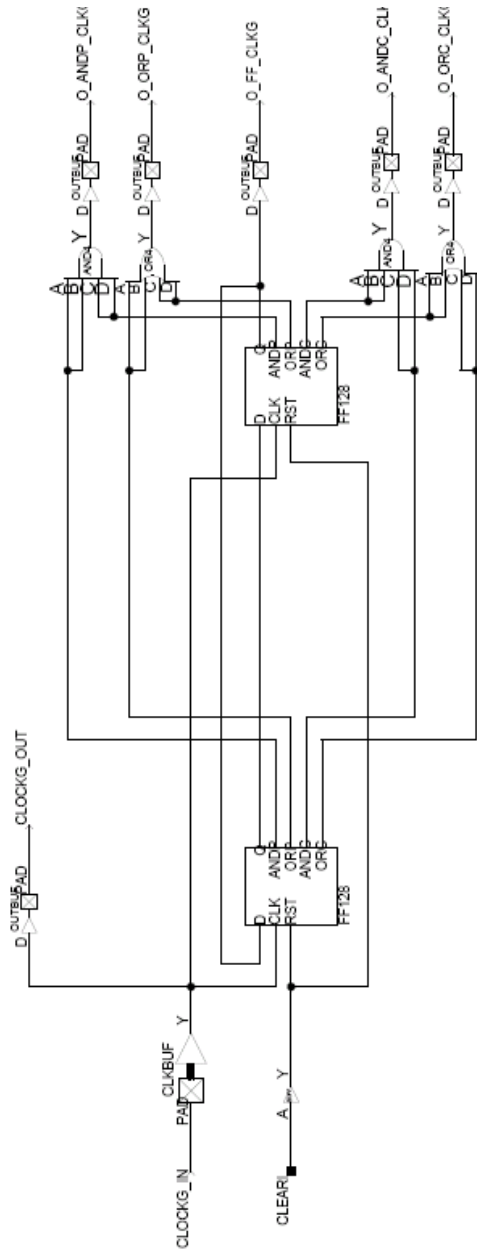


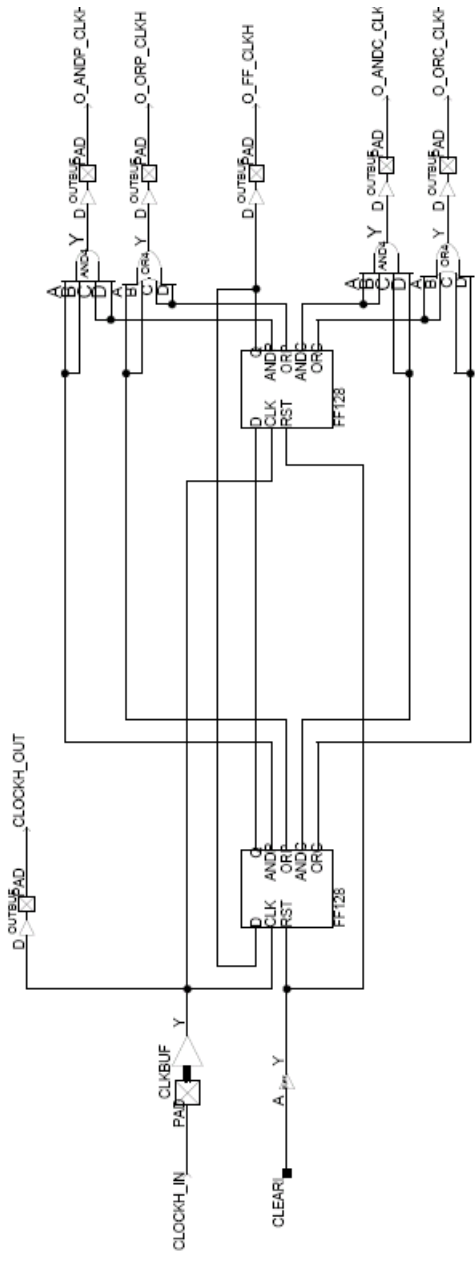












```

// 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*/;
```

```
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(anc_a), .B(anc_b), .C(anc_c), .D(anc_d), .Y(ANDC));
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 anc_a, anc_b, anc_c, anc_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(anc_a), .ORC(orc_a));

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

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

FF8 dff_d (.D(x3), .Q(Q), .CLK(CLK), .RST(RST), .ANDP(andp_d), .ORP(orp_d),
.ANDC(anc_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(anc_a), .B(anc_b), .C(anc_c), .D(anc_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));

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,WClck,RClock);
input [5:0] Data;
output [5:0] Q;
input [10:0] WAddress, RAddress;
input WE, RE, WClck, RClock;

  wire WEP, REP, VCC, GND;

  VCC VCC_1_net(.Y(VCC));
  GND GND_1_net(.Y(GND));
  RAM64K36P ram_2048x6_R0C2(.WCLK(WClck), .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
```