Lab3

实验目的

1. 实现一个组相连的 cache
2. 用新的 cache 替换原有 cache，使用快速排序和矩阵乘法

实验环境与工具

Windows 环境下使用 vivado 仿真和综合工具

实验内容与过程

阶段一：实现 LRU 策略与 FIFO 策略的组相连
开始时，按照 WAY_CNT 的顺序写入。FIFO 只需记录每一组的第一路，以后每次都替换下一路，替换的路为下图所示

```
index[set_addr] <= (index[set_addr]+1) % WAY_CNT;
```

LRU 需要用一个数组记录一组中每一路的距离上次使用的时间间隔，0 为最近使用，WAY_CNT-1 表示最久未使用。每次命中就把对应路的使用间隔变为 0，并把使用间隔低于其的路使用间隔都变大 1 时。每次替换使用间隔最大的路
代码如下
way_cnt 为命中的路
Cache 命中后，使用间隔低于命中路的间隔加一，命中路间隔为 0

```
if(cache_hit) begin
    for(integer e = 0 ; e < WAY_CNT ;e++) begin
        if(lru_decision[set_addr][e] < lru_decision[set_addr][way_cnt] )
            lru_decision [set_addr][e] = lru_decision[set_addr][e] + 1;
    end
    lru_decision[set_addr][way_cnt] <= 0;
```

Cache miss 后，找到使用间隔最大的路，记为 index 用于换入换出。最后把所有的使用间隔都加 1，由于取余，原本间隔最大的正好变为 0。

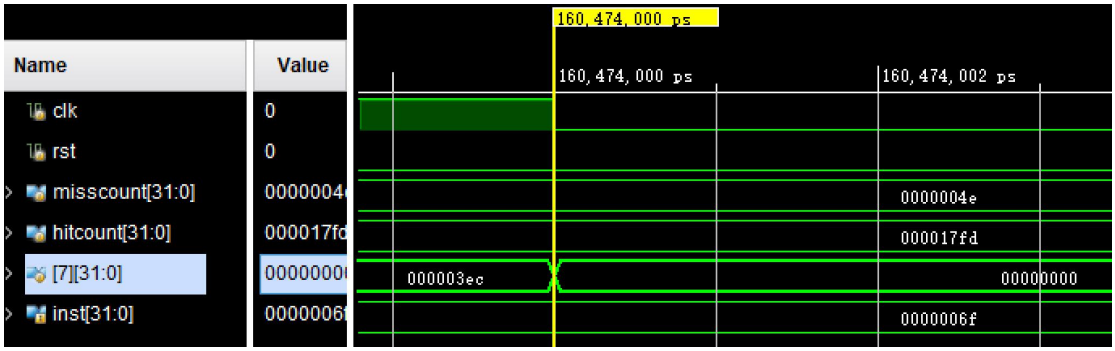
```
for(integer e = 0 ; e < WAY_CNT ;e++) begin
    if(lru_decision[set_addr][e] == WAY_CNT - 1 ) begin
        index = e;
        break;
    end
end
```

```
begin          // 上一个周期换入成功，这周期将主存读出的line写入cache，并更新tag，置高valid,
    for(integer e = 0 ; e < WAY_CNT ;e++)
        lru_decision [set_addr][e] = (lru_decision[set_addr][e] + 1) % WAY_CNT;
    for(integer i=0; i<LINE_SIZE; i++)  cache_mem[mem_rd_set_addr][index][i] <= mem_rd_
```

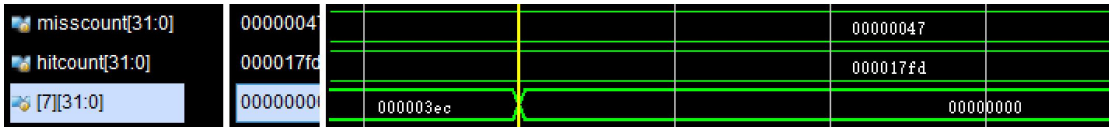
在默认设定中

```
module cache #(
    parameter  LINE_ADDR_LEN = 3, // 
    parameter  SET_ADDR_LEN  = 3, // 
    parameter  TAG_ADDR_LEN  = 6, // 
    parameter  WAY_CNT       = 3  // 
)(
```

FIFO 快排缺失率为



LRU 快排缺失率为



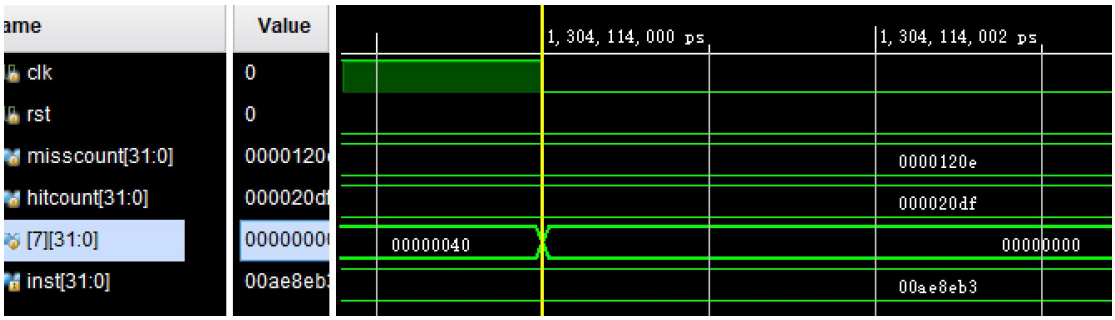
可以看到两种策略的缺失率几乎没有差别。
由于原本的数据中 cache 里有 8*8*3=192 个 word，而快速排序中只有 255 个元素，所以在默认数据时，FIFO 与 LRU 的差距并不明显。
所以在下面改变参数时我们使用矩阵乘法的结果进行比较

开始执行指令的时间都是 12000ps
使用测试程序为默认的矩阵乘法

如下设置

```
module cache #(
    parameter  LINE_ADDR_LEN = 3, // 
    parameter  SET_ADDR_LEN  = 3, // 
    parameter  TAG_ADDR_LEN  = 6, // 
    parameter  WAY_CNT       = 3  // 
)(
```

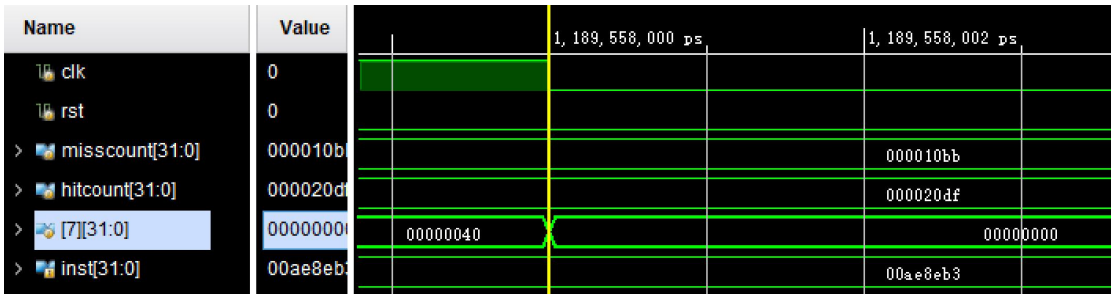
FIFO 缺失率如下图



综合结果如下图

	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP	Start	Elapsed	Run Strategy
							3297	7293	4.00	0	0	5/1...	00:01:11	Vivado Synthesis Def
	NA	NA	NA	NA	11.464	0	3297	7293	4.00	0	0	5/1...	00:01:51	Vivado Implementatio

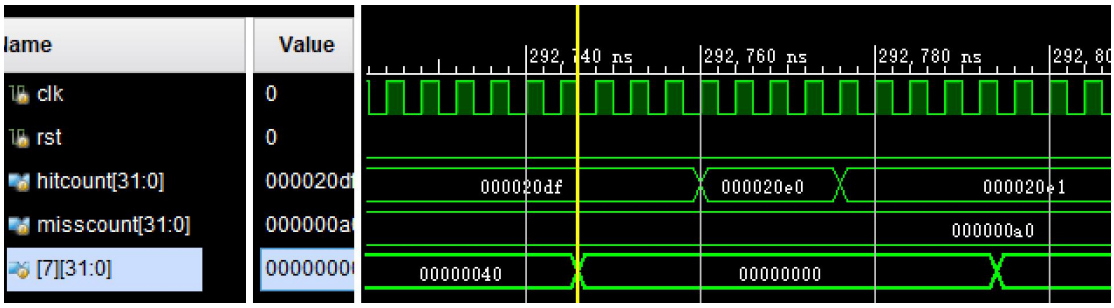
LRU 缺失率如下



	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	BRAMs	FF	URAM	DSP	Start
								3279	4.00	7375	0	0	5/16/20
	NA	NA	NA	NA	NA	9.980	0	3277	4.00	7375	0	0	5/16/20

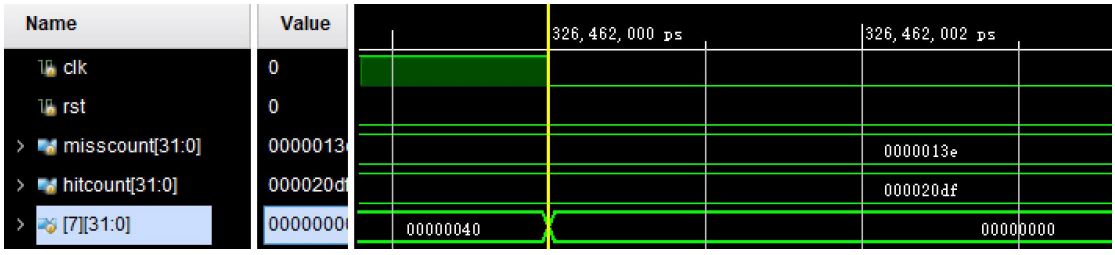
```
module DataCache #(
    parameter  LINE_ADDR_LEN = 3, //
    parameter  SET_ADDR_LEN  = 3, //
    parameter  TAG_ADDR_LEN  = 6, //
    parameter  WAY_CNT       = 6  //
)(
```

FIFO：



TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM
						5920	13637	4.00	0
NA	NA	NA	NA	20.258	0	5920	13637	4.00	0

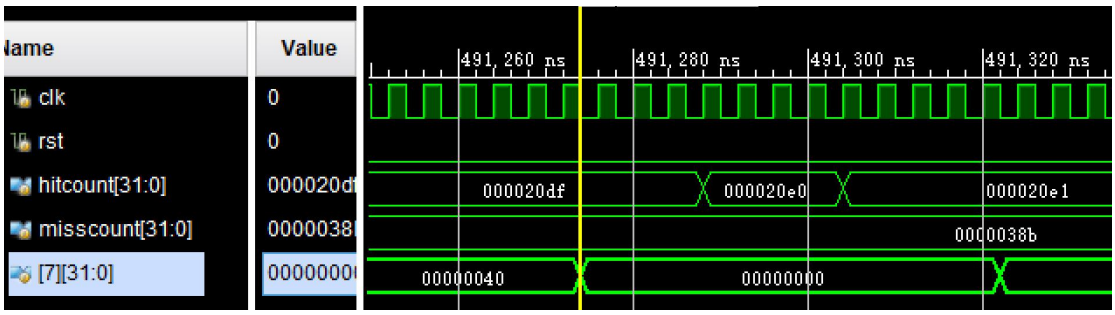
LRU：



WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	BRAMs	FF	URAM	DSP	Start
							6861	4.00	13952	0	0	5/16/20 5:46 PM
NA	NA	NA	NA	NA	29.661	0	6859	4.00	13952	0	0	5/16/20 5:48 PM

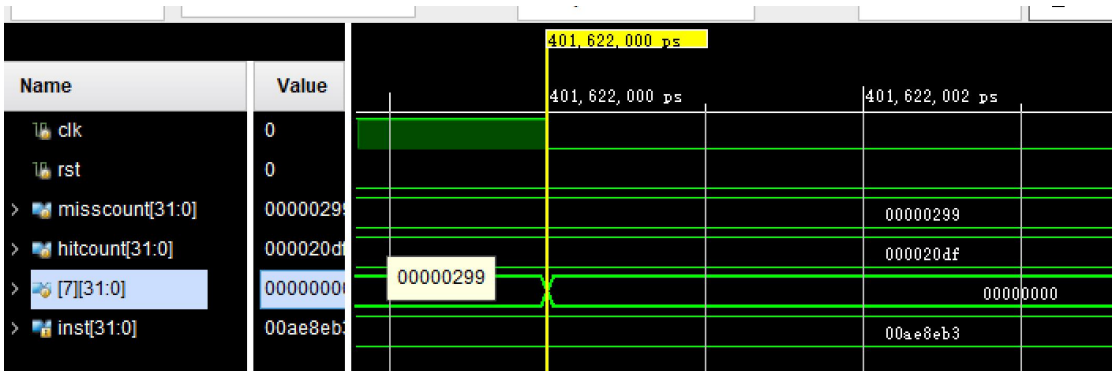
```
module DataCache #(
    parameter LINE_ADDR_LEN = 3, //
    parameter SET_ADDR_LEN  = 4, //
    parameter TAG_ADDR_LEN  = 6, //
    parameter WAY_CNT       = 3  //
)(
```

FIFO：



THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP
				5374	13649	8.00	0	0
NA	NA	22.267	0	5373	13649	8.00	0	0

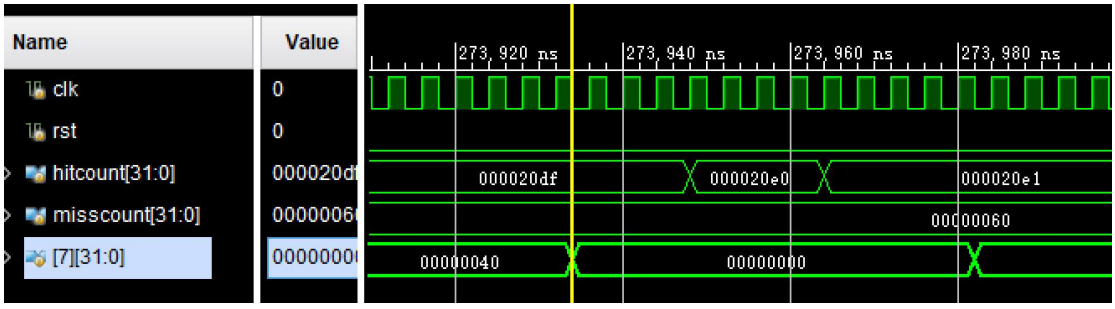
LRU:



WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	BRAMs	FF	URAM	DSP
							5962	8.00	13811	0	0
NA	NA	NA	NA	NA	22.321	0	5959	8.00	13811	0	0

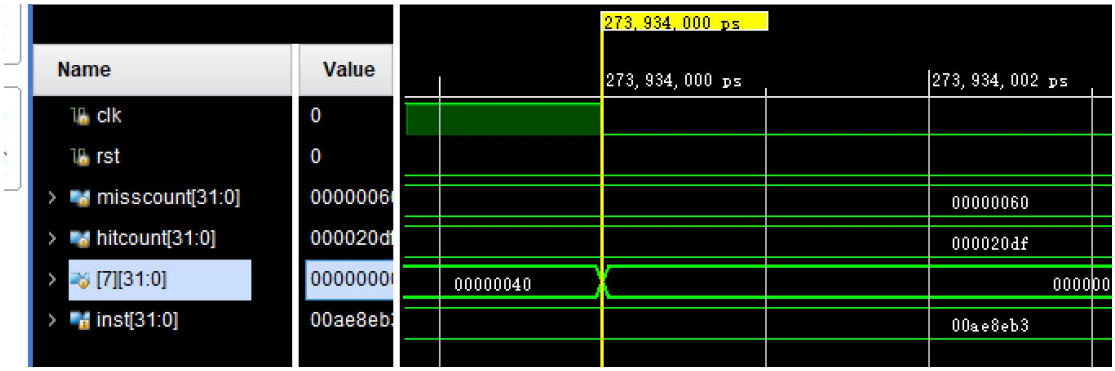
```
module DataCache #(
    parameter LINE_ADDR_LEN = 3,
    parameter SET_ADDR_LEN  = 4,
    parameter TAG_ADDR_LEN  = 6,
    parameter WAY_CNT       = 6
)(
```

FIFO：



TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP
						9228	26337	8.00	0	0
NA	NA	NA	NA	40.149	0	9221	26337	8.00	0	0

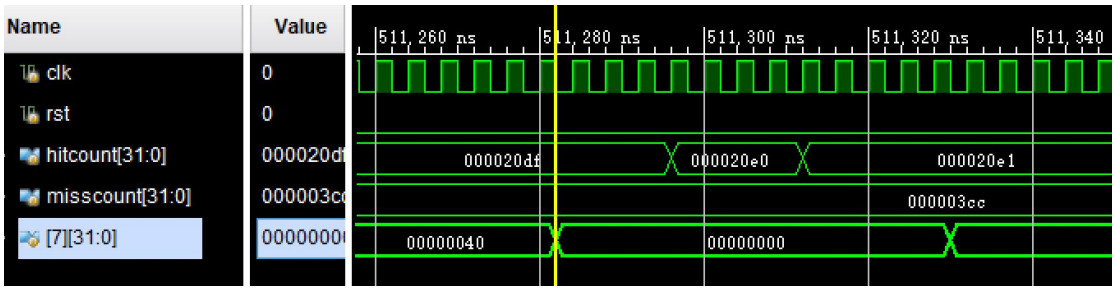
LRU：



	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	BRAMs	FF	URAM	DSP	Start
								11355	8.00	26964	0	0	5/16/2
	NA	NA	NA	NA	NA	37.794	0	11354	8.00	26964	0	0	5/16/2

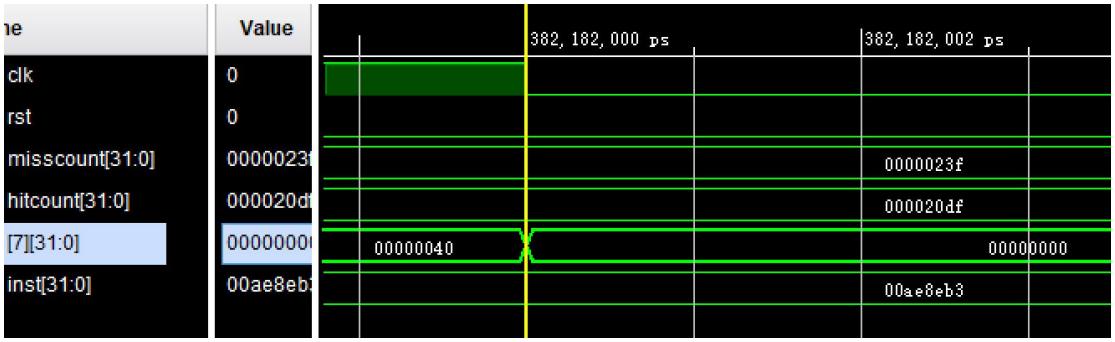
```
module DataCache #(
    parameter  LINE_ADDR_LEN = 4, // li
    parameter  SET_ADDR_LEN  = 3, // 组
    parameter  TAG_ADDR_LEN  = 6, // ta
    parameter  WAY_CNT       = 3  // 组
```

FIFO:



S	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP	Start	Elapsed	Run
							5934	14206	8.00	0	0	5/15/20 9:51 PM	00:01:18	Viva
	NA	NA	NA	NA	22.620	0	5930	14206	8.00	0	0	5/15/20 9:52 PM	00:02:27	Viva

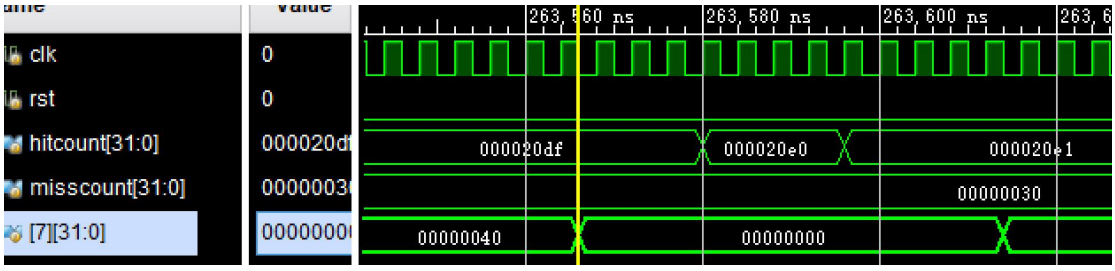
LRU:



WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	BRAMs	FF	URAM	DSP	Start
							6279	8.00	14288	0	0	5/16/20 7:
NA	NA	NA	NA	NA	22.664	0	6277	8.00	14288	0	0	5/16/20 7:

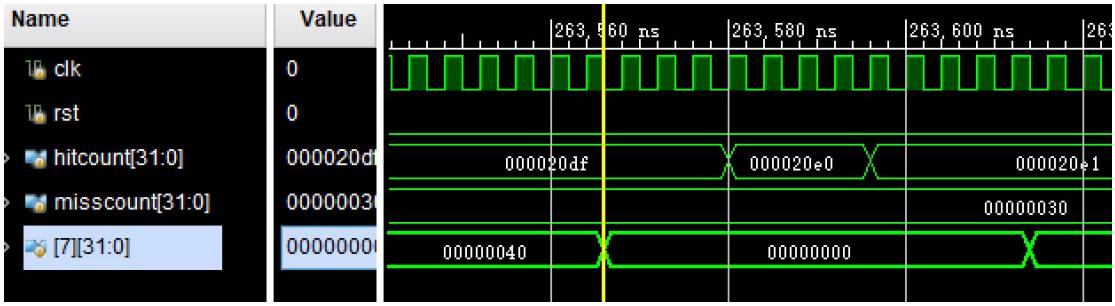
```
module cache #(
    parameter  LINE_ADDR_LEN = 4, // li
    parameter  SET_ADDR_LEN  = 3, // 组
    parameter  TAG_ADDR_LEN  = 6, // ta
    parameter  WAY_CNT       = 6  // 组
```

FIFO：



WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	BRAMs	FF	URAM	DSP
							11672	8.00	26694	0	0
NA	NA	NA	NA	NA	44.980	0	11668	8.00	26694	0	0

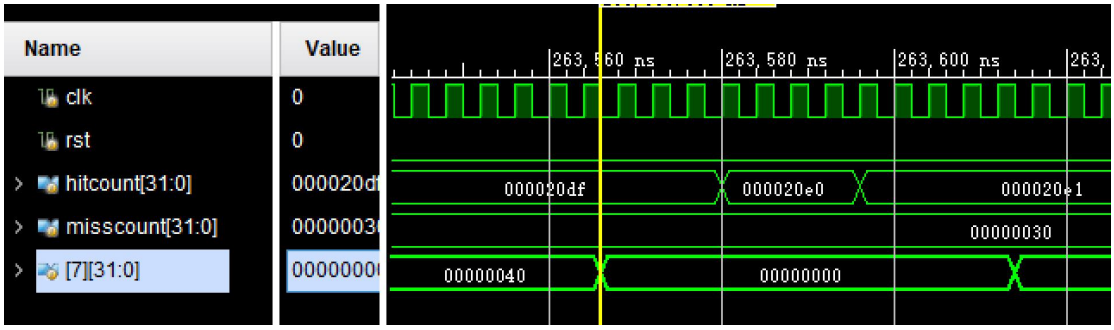
LRU：



WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	BRAMs	FF	URAM	DSP
							11439	8.00	27009	0	0
NA	NA	NA	NA	NA	39.364	0	11432	8.00	27009	0	0

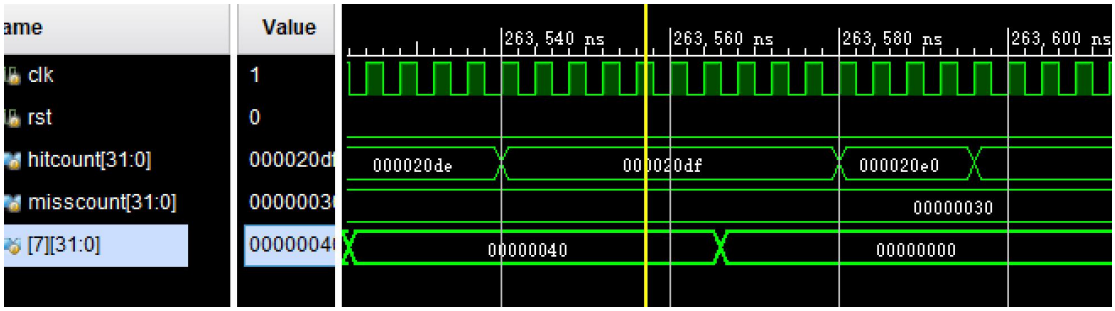
```
    parameter  LINE_ADDR_LEN = 4, //
    parameter  SET_ADDR_LEN  = 4, //
    parameter  TAG_ADDR_LEN  = 6, //
    parameter  WAY_CNT       = 3  //
)(
```

FIFO：



	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	BRAMs	FF	URAM
								10010	16.00	26712	0
	NA	NA	NA	NA	NA	46.296	0	10003	16.00	26712	0

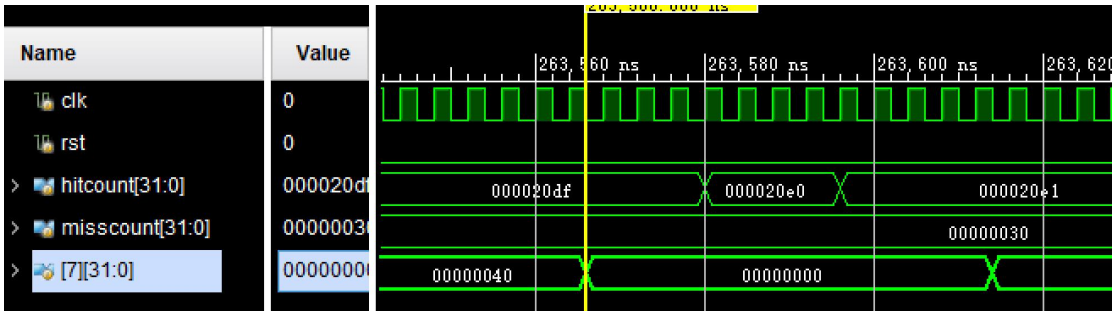
LRU：



NS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	BRAMs	FF	URAM	DSP
							9818	16.00	26874	0	0
A	NA	NA	NA	NA	40.308	0	9803	16.00	26874	0	0

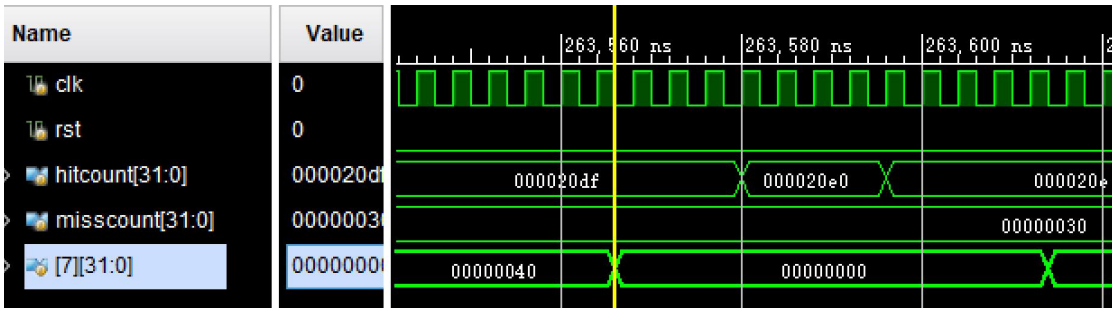
```
module cache #(
    parameter LINE_ADDR_LEN = 4, // 
    parameter SET_ADDR_LEN  = 4, // 
    parameter TAG_ADDR_LEN  = 6, // 
    parameter WAY_CNT       = 6  // 
)(
```

FIFO：



WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	BRAMs	FF	URAM	DSP	Start
							17955	16.00	51688	0	0	5/16/2
NA	NA	NA	NA	NA	81.689	0	17926	16.00	51688	0	0	5/16/2

LRU:



WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	BRAMs	FF	URAM	DSP
							20574	16.00	52319	0	
NA	NA	NA	NA	NA	93.645	0	20565	16.00	52319	0	

我们设开始时间 12000ps（较小忽略），结束时间为七号寄存器变为 0 的时间。时间单位为 100000ps

电路面积用 total power 表示，以 10000 为单位 1

以时间反比

将数据统计为表格得

替换策略	LINE_ADDR_LEN	SET_ADDR_LEN	WAY_CNT	缺失率	运行时间	电路面积	效率与面积之比 *100
FIFO	3	3	3	54.9%	13.0	1.14	6.75
LRU	3	3	3	50.9%	11.9	1.00	8.40
FIFO	3	3	6	1.90%	2.92	2.02	17.0
LRU	3	3	6	3.78%	3.26	2.97	10.3
FIFO	3	4	3	10.8%	4.91	2.23	9.13
LRU	3	4	3	7.90%	4.01	2.23	11.2
FIFO	3	4	6	1.14%	2.74	4.01	9.10
LRU	3	4	6	1.14%	2.74	3.78	9.66
FIFO	4	3	3	11.5%	5.11	2.26	8.66
LRU	4	3	3	6.83%	3.82	2.27	11.5
FIFO	4	3	6	0.57%	2.64	4.50	8.42
LRU	4	3	6	0.57%	2.64	3.94	9.61
FIFO	4	4	3	0.57%	2.64	4.63	8.18
LRU	4	4	3	0.57%	2.64	4.03	9.40
FIFO	4	4	6	0.57%	2.64	8.17	4.64
LRU	4	4	6	0.57%	2.64	9.37	4.04

由表格数据容易发现：

1. 总体来说 cache 电路的面积和 cache 的容量成正比
2. 在 cache 容量没有接近数据规模的情况下，
3. 当 cache 较小时，缺失率很高，需要大量的换入换出
4. 通过变大 LINE_ADDR_LEN，SET_ADDR_LEN，WAY_CNT 使 cache 容量翻倍，都可以使缺失率和运行时间显著减小，其中 WAY_CNT 增大降低缺失率的效果最好。
5. 通过效率与面积之比，我们得出第三组数据整体功耗最好。
6. 当 cache 容量接接近或超过数据规模的时候，两种策略效果几乎一样，因为这个时候只需将所有数据换入 cache 即可，增大 cache 在缺失率与运行时间上提升并不显著。

实验总结

本次实验难点主要在 LRU 的实现上，要充分理解 LRU 的原理才能在硬件上较好的实现。
本次实验的一个易错点是记录缺失率的时候，命中次数与缺失次数都要在有读写请求的时候才能增加，否则会在 cache 空闲的时候不停自增。
通过本次实验，我初步掌握了 FIFO 与 LRU 两种替换策略以及 cache 参数变化时对性能的影响。

意见

无