

9.24

芯片像素阵列延迟模型

高能物理研究所

郁晗



目录

- 更新不同信号的延迟仿真结果
 - RC提取使用更符合实际的情况（驱动电阻未优化）
 - 延时>>预估驱动电阻
 - 时序分析图



更新参数后各信号延迟估算

原 16um: $R = 35 \Omega$ $C = 3 \text{ fF}$ 30um: $R = 50 \Omega$ $C = 4 \text{ fF}$
 使用更符合实际情况的RC仿真值 16um: $R = 12 \Omega$ $C = 4 \text{ fF}$ 30um: $R = 27 \Omega$ $C = 5.2 \text{ fF}$ (预估)

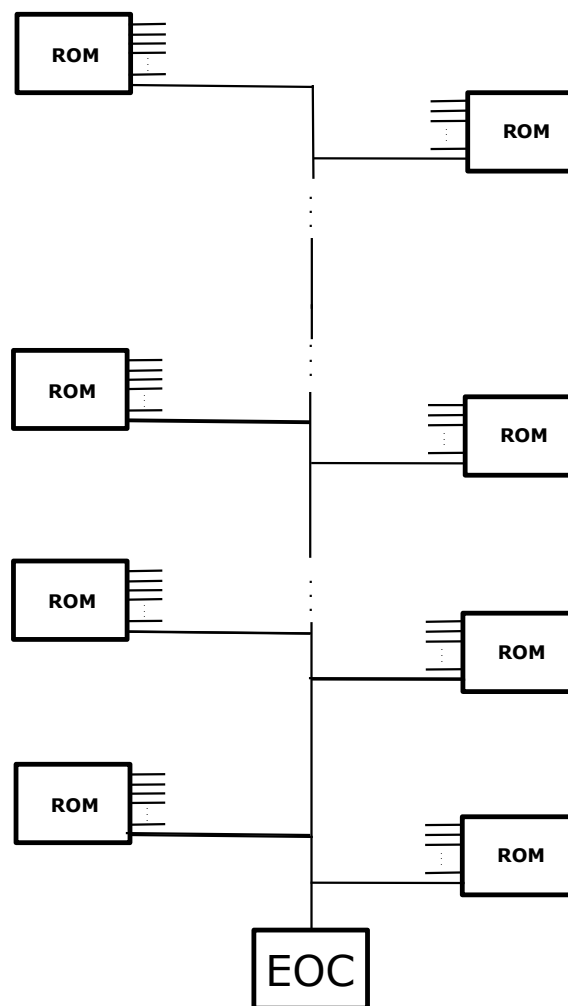
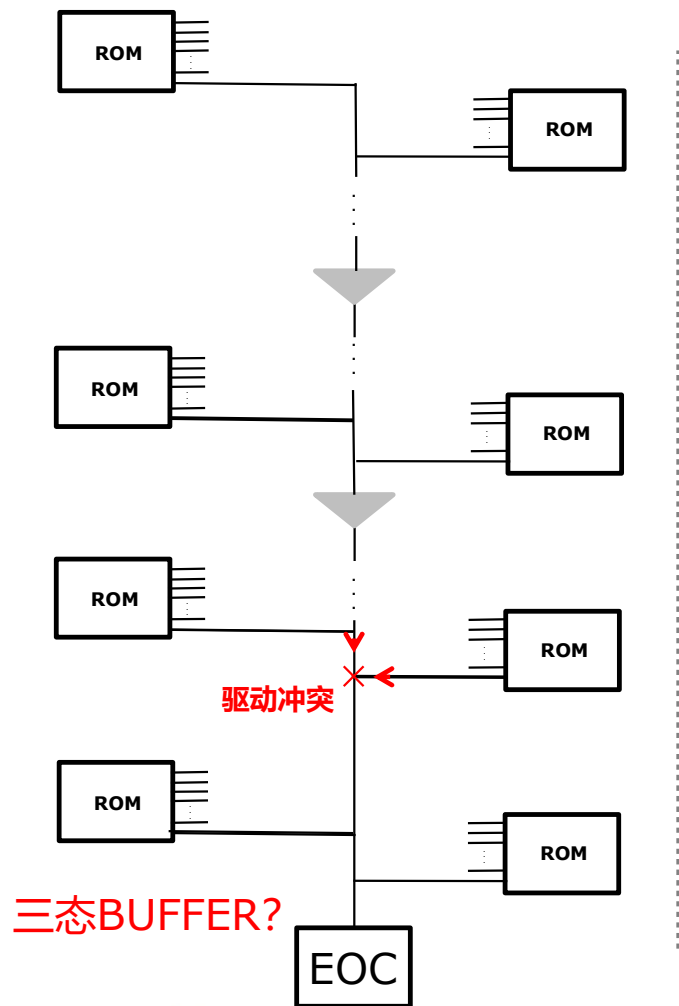
信号说明		Pixel to pixel NOR ns	Pixel to pixel NAND ns	Group ns
Slow Hit	SET	0.1237 >> 0.1236	0.1590 >> 0.1589	4.5233 >> 4.5218
	RESET	0.3158 >> 0.3157	0.1977 >> 0.1976	8.2161 >> 8.2146
信号说明		Group ns		1024 Pixels ns
Fastor	RISE	2.3836 >> 2.9921		76.2744 >> 95.7485
	FALL	1.2511 >> 1.4702		40.0372 >> 47.0475
Read /Freeze	RISE	3.0692 >> 3.6397		98.1901 >> 117.1896
	FALL	1.3459 >> 1.5409		43.0604 >> 49.5961

慢链 优化可忽略不计
没有驱动电阻的影响

快链 和 READ/FREEZE 延迟 增加
存在驱动电阻的影响 且寄生电容增大

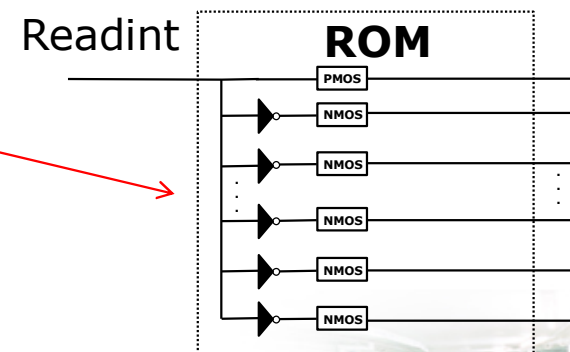


ADDR考虑



- 对于ADDR来说，加buffer会造成驱动冲突>>无法添加buffer

- 需要对关注较远ROM延时

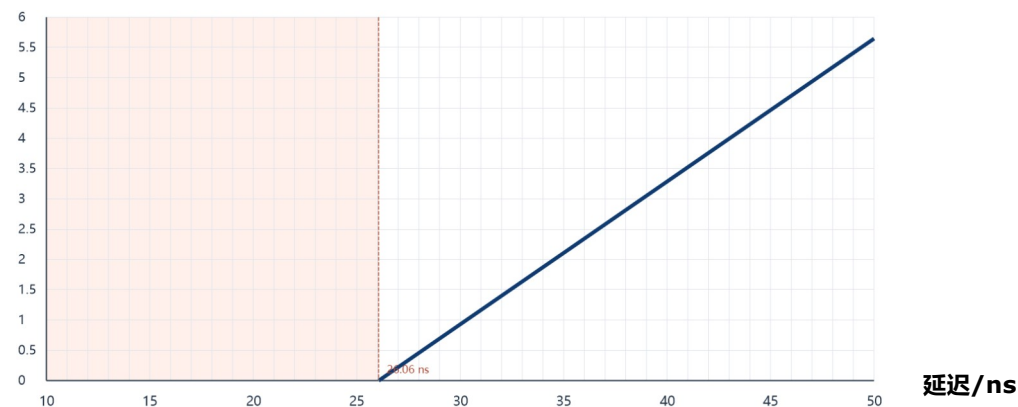


ADDR

延时>>驱动电阻

ADDR	
延迟	预估驱动电阻
10ns	无法达到
20ns	无法达到
26.06 ns	0Ω (理想)
30ns	0.913KΩ
40ns	3.289KΩ
50ns	5.648KΩ

预估驱动电阻/kΩ



ADDR 延迟和预估驱动电阻关系

考虑到**ROM面积**问题，结合**时序余量**
48 ns 是一个比较有希望的值，去尽力优化
(5.2KΩ)

延迟/ns

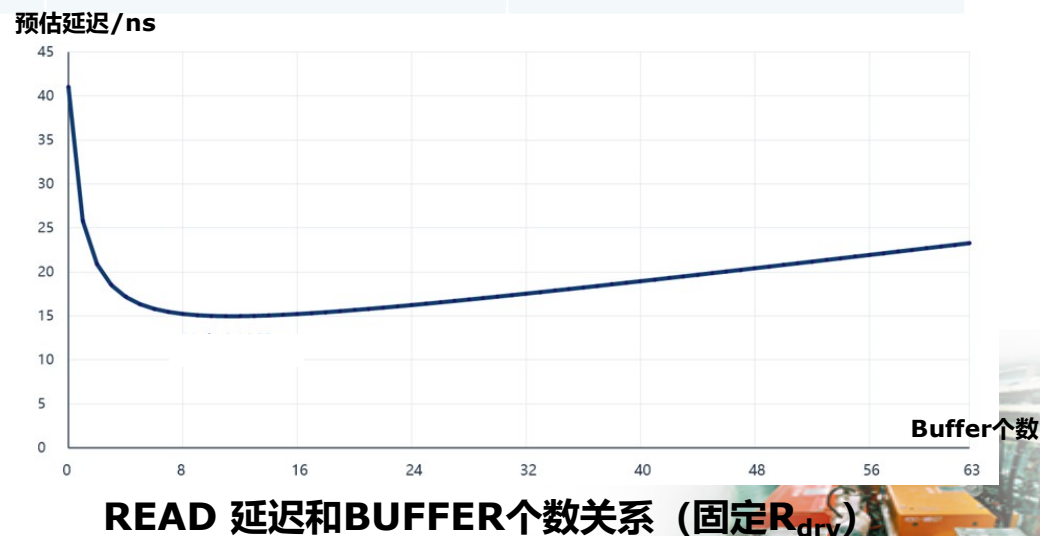


Read

延时>>驱动电阻

READ (驱动电阻 2K Ω)				
中继BUFFER个数	门延迟 ns	线延迟 ns	驱动充电延迟 ns	总延迟时间 ns
0	0.1976	30.7917	10.0332	41.0225
1	0.3951	15.3876	10.0354	25.8181
3	0.7903	7.6855	10.0398	18.5156
8	1.5806	3.8345	10.0486	15.4637
15	3.1611	1.9090	10.0663	15.1364
31	6.3222	0.9462	10.1016	17.3701
63	12.6445	0.4648	10.1723	23.2816

READ (驱动电阻 20K Ω) 对比				
中继BUFFER个数	门延迟 ns	线延迟 ns	驱动充电延迟 ns	总延迟时间 ns
0	0.1976	30.7917	100.3315	131.3208
1	0.3951	15.3876	100.3536	116.1363
3	0.7903	7.6855	100.3978	108.8736
8	1.5806	3.8345	100.4861	105.9011
15	3.1611	1.9090	100.6627	105.7328
31	6.3222	0.9462	101.0160	108.2845
63	12.6445	0.4648	101.7226	114.8319



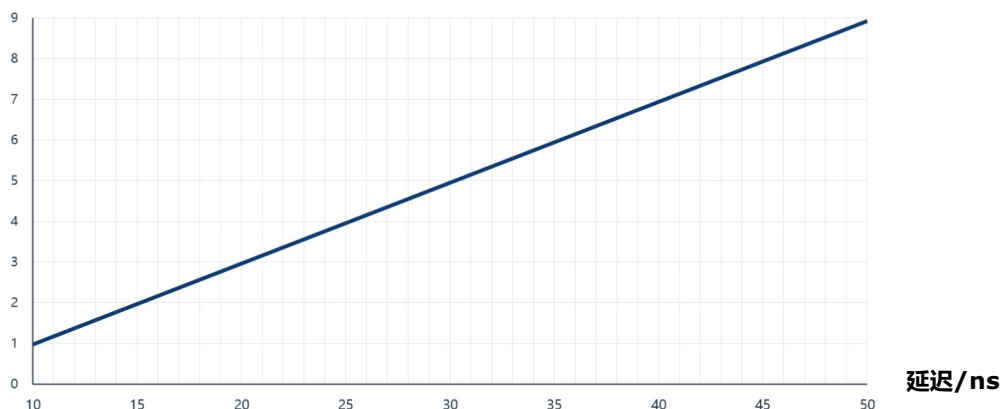
Read

延时>>驱动电阻

Read 总延时目标	固定门延时累计	线 RC 延时累计	扣除门+线延时后的驱动项预算	各级上拉电阻上限
10 ns	3.1611 ns	1.9090 ns	4.9299 ns	0.9795 kΩ
20 ns	3.1611 ns	1.9090 ns	14.9299 ns	2.9663 kΩ
30 ns	3.1611 ns	1.9090 ns	24.9299 ns	4.9532 kΩ
40 ns	3.1611 ns	1.9090 ns	34.9299 ns	6.9400 kΩ
50 ns	3.1611 ns	1.9090 ns	44.9299 ns	8.9268 kΩ

10ns → 1kΩ
30ns → 5kΩ

预估驱动电阻/kΩ



READ 延迟和预估驱动电阻关系

较好评估是可以达到10ns
保守评估是可以达到30ns

考虑到Buffer面积和时序余量可以选择32ns



Freeze

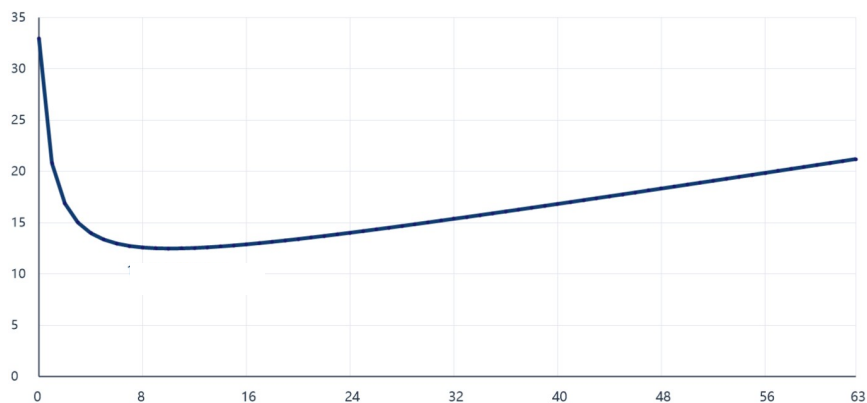
延时>>驱动电阻

FREEZE (驱动电阻 2 k Ω)

中继 BUFFER 个数	门延迟 ns	线延迟 ns	驱动充电延迟 ns	总延迟时间 ns
0	0.1976	24.7201	8.0548	32.9725
1	0.3951	12.3548	8.0570	20.8069
3	0.7903	6.1721	8.0614	15.0238
7	1.5806	3.0807	8.0702	12.7315
15	3.1611	1.5351	8.0879	12.7841
31	6.3222	0.7622	8.1232	15.2077
63	12.6445	0.3758	8.1939	21.2142

不同中继Buffer数对延时的影响

预估延迟/ns



Buffer个数

Freeze 到最远 Pixel0 的目标延时	固定门+线 RC 延时	驱动充电延时累计预算	各级上拉驱动电阻上限
10 ns	4.6613 ns	5.3387 ns	1.3231 k Ω
20 ns	4.6613 ns	15.3387 ns	3.8013 k Ω
30 ns	4.6613 ns	25.3387 ns	6.2795 k Ω
40 ns	4.6613 ns	35.3387 ns	8.7578 k Ω
50 ns	4.6613 ns	45.3387 ns	11.2360 k Ω

延迟做到10ns, R_{drv} 需要低至1.3K Ω
延迟做到30ns, R_{drv} 需要低至 6 K Ω

Hit_or / Fast_or

延时>>驱动电阻

分组数	每组像素数	本组慢链清零	整条快链清零	慢链+快链清零
2	512	131.4334 ns	12.1754 ns	143.6088 ns
4	256	65.7167 ns	7.4272 ns	73.1439 ns
8	128	32.8583 ns	6.3961 ns	39.2544 ns
16	64	16.4292 ns	8.5663 ns	24.9955 ns
32	32	8.2146 ns	15.0231 ns	23.2377 ns
64	16	4.1073 ns	28.9949 ns	33.1022 ns

分组数	每组像素数	本组慢链建立	整条快链建立	慢链+快链建立
2	512	72.3486 ns	12.0719 ns	84.4205 ns
4	256	36.1743 ns	7.2204 ns	43.3947 ns
8	128	18.0871 ns	5.9824 ns	24.0695 ns
16	64	9.0436 ns	7.7390 ns	16.7825 ns
32	32	4.5218 ns	13.3684 ns	17.8902 ns
64	16	2.2609 ns	25.6855 ns	27.9463 ns

16Group与32Group差别不大
先保留32Group结构

快链驱动电阻 kΩ	慢链建立 ns	慢链清零 ns	快链建立 ns	快链清零 ns	慢链+快链建立 ns	慢链+快链清零 ns
20	4.5218	8.2146	87.6455	89.3002	92.1673	97.5148
8	4.5218	8.2146	43.0793	44.7340	47.6010	52.9486
4	4.5218	8.2146	28.2238	29.8786	32.7456	38.0931
2	4.5218	8.2146	20.7961	22.4508	25.3179	30.6654
1	4.5218	8.2146	17.0823	18.7370	21.6040	26.9516
0 (理想参考)	4.5218	8.2146	13.3684	15.0231	17.8902	23.2377

快链或门驱动降到2kΩ
总延迟预计来到30ns

快链或门驱动降到8kΩ
总延迟预计来到52ns

选择总延迟为48ns
此时驱动大概为7KΩ

较好估计 (追求速度)

Read + Fastor&Hitor
10ns 18ns 8ns
36ns

Read + Addr
10ns 30ns
40ns

保守估计 (兼顾速度和面积)

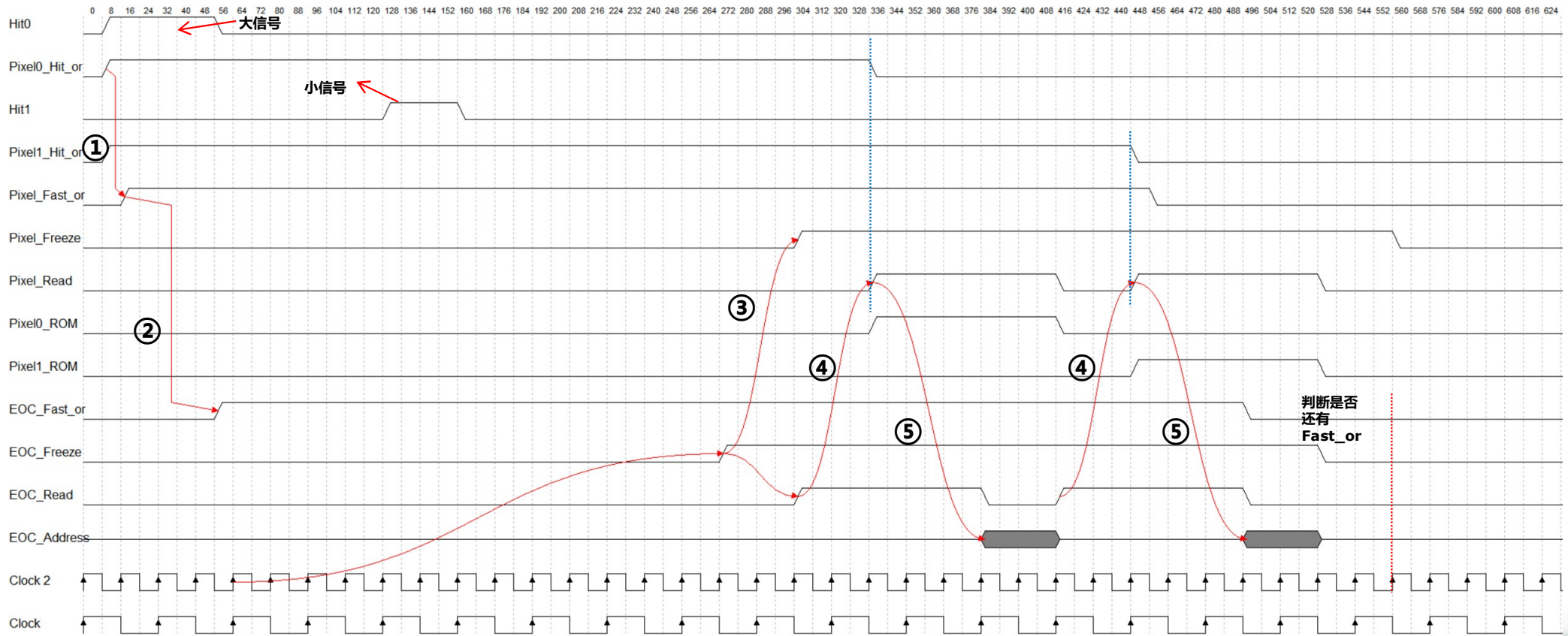
Read + Addr
32ns 48ns
80ns

Read + Fastor&Hitor
32ns 40ns 8ns
80ns

基本满足目标



时序分析图



实际周期: clock2 1/128Mhz (7.8125ns)

①Hit_or 延迟: 8ns

②Fast_or 延迟: 40ns

③Freeze 延迟: 32ns

④Read 延迟: 32ns

为了形象这里使用: 1/125Mhz (8ns) 代替

⑤ADDR 延迟: 48ns



总结

- 选择 Addr **48 ns** Read / Freeze **32 ns** Hit_or / Fast_or **8 / 40 ns**

评估 driver 的 MOS 管尺寸和电阻, 延迟时间是否大致符合

ADDR

C_{rom}	=	2	ff (假定, 实际上在调大mos管尺寸的同时, C也会变大, 所以这里会存在一个不确定性)
C_{wire}	=	4	ff
C_{rev}	=	4	ff
: $1023 * C_{wire} + 1023 * C_{rom} + C_{rev}$			

READ 16段

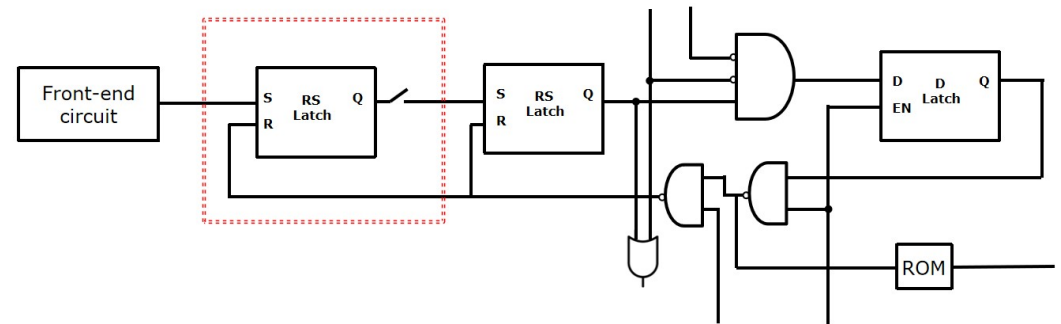
$C_{PIXEL_INPUT_FF}$	=	1.8+1.3	ff
C_{wire}	=	4	ff
$C_{BUFFER_IN_FF}$	=	1.6	ff
: $64 * C_{PIXEL_INPUT_FF} + 63 * C_{wire} + C_{BUFFER_IN_FF}$			

- 初步确定像素击中时序图
- 后续根据实际阵列架构, 继续修改行为级延迟模型



backup

- 选择addr 30ns
- 选择read 20ns
- 评估driver的MOS管尺寸和电阻



ADDR

$C_{rom} = 2 \text{ ff}$ (假定, 实际上在调大mos管尺寸的同时, C也会变大, 所以这里会存在一个不确定性)

$C_{wire} = 4 \text{ ff}$

$C_{rev} = 4 \text{ ff}$

$1023 * C_{wire} + 1023 * C_{rom} + C_{rev}$

READ 16段

$C_{PIXEL_INPUT_FF} = 1.8 + 1.3 \text{ fF}$

$C_{wire} = 4 \text{ ff}$

$C_{BUFFER_IN_FF} = 1.6 \text{ fF}$

$64 * C_{PIXEL_INPUT_FF} + 63 * C_{wire} + C_{BUFFER_IN_FF}$

