doi:10.3772/j.issn.1002-0470.2018.02.001

# 基于自适应时序匹配的低延迟寄存器堆①

元国军②[*,**] 沈 华[*] 邵 恩[*,**] 臧大伟[*]

([*]中国科学院计算技术研究所 北京 100190)

([**]中国科学院大学计算机与控制学院 北京 100049)

**摘 要** 指出半导体工艺与晶体管特性参数的随机波动随着芯片特征尺寸不断减小越来越大,传统的基于预匹配的寄存器堆设计方法必须通过增大匹配裕量来保证读写操作的可靠性,为了克服制约寄存器堆性能提升的这一关键因素,提出了一种基于自适应时序匹配的低功耗寄存器堆电路结构。该结构通过对多端口寄存器堆的访存时序进行自适应匹配与调优,达到减小寄存器堆访问延时、降低功耗以及提高芯片工艺敏感度的目的。电路及版图仿真结果显示:基于该方法实现的3 读2 写 32 ×64 bit 寄存器堆,在 SMIC 40nm 工艺条件下,芯片面积为135.5μm ×65.1μm,访存延迟为 357ps,相比于传统的 Chain Delay 匹配技术,延迟减小 22%,功耗降低 35%。

**关键词** 多端口寄存器堆,自适应时序匹配,低延迟,低功耗,静态随机存储器

## 0 引 言

多端口寄存器堆是多核和众核处理器的重要组成模块,它可在一个时钟周期内同时为多个运算部件提供数据存储和读写服务,其性能优劣直接影响处理器的时序特性、功耗、面积及稳定性[1,2]。访存延时是多端口寄存器堆的关键性能参数之一,直接决定寄存器堆的工作频率大小[3]。访存延迟大小与寄存器的堆使能控制时序密切相关,为了改进寄存器堆的访存性能与功耗,需要精确控制灵敏放大器使能时序[4]。传统的时序控制策略是在灵敏放大器的使能控制路径上构造出一条与寄存器堆数据通路延时精确匹配的延时链,通过匹配延迟链的变化来调整灵敏放大器的使能信号,减小访存延迟,如文献[5]提出的 Chain Delay 和 Replica 方法。

随着集成电路进入纳米时代,半导体工艺与晶体管特性的随机离散性随线宽缩小而不断增大,因此越来越难以在电路设计与芯片实际运行之间实现预先设定的匹配精度,在设计过程中进行预匹配、增加时序裕量的传统方法,在实际中难以同时满足低功耗、高性能、合理的成品率与成本等多种要求[6,7]。如何有效克服这类挑战是设计与实现低功耗、高性能寄存器堆的关键技术之一。针对这一问题,本文提出了一种支持自适应时序匹配的低延迟寄存器堆结构,它基于自测试、可调节的延迟单元的自适应延时补偿机制,可更灵活地实现寄存器堆灵敏放大器使能的延时匹配与控制,跟踪与补偿多种因素变化下寄存器堆灵敏放大器控制使能的时钟产生与脉冲宽度,降低寄存器堆电路访问延迟中的冗余量,减小访存延迟;同时自适应匹配还有效减少了寄存器堆中耗电部件的开启时间,在完成访存操作后及时关闭相关模块,降低了访存功耗。本文的主要贡献是给出了基于自适应时序匹配的多端口寄存器堆结构和各模块实现方法,阐述了时序反馈机制和匹配机制,完成了 3 读 2 写多端口寄存器堆的电

① 国家自然科学基金(61572464,61331008)和十三五国家重点研发计划(2016YFB0200205)资助项目。

② 男,1983 年生,博士生,高级工程师;研究方向:计算机系统结构,集成电路设计,光互连网络等;联系人,E-mail: yuanguojun@ncic.ac.cn

(收稿日期:2017-08-18)

路及版图设计。实验显示,基于该方法实现的自适应时序多端口寄存器堆,在 SMIC 40nm 工艺条件下,相比于 Chain Delay 匹配技术的寄存器堆结构,延迟减小 22%,功耗降低 35%,自适应模块额外的面积开销仅占总面积的 1%。

## 1　自适应时序匹配的寄存器堆结构

如图 1 所示,寄存器堆主要由存储阵列(SRAM Array)、自适应时序控制(adjustable timing controller)、灵敏放大器(sense amplifier)、预充电(precharge)电路、译码器(decoder)和驱动电路(driver)等组成。其中存储阵列基于静态随机存储器(static random access memory, SRAM)[8,9]实现,负责所有数据的存储功能;行译码器(row decoder)和字线驱动器(wordline driver)模块提供读写的行地址信号;sense amplifier、write driver 和 output driver 等模块完成数据的读写功能;adjustable timing controller 可以自适应匹配实际芯片的路径时序,通过控制 sense amplifier 来调整访存时序。

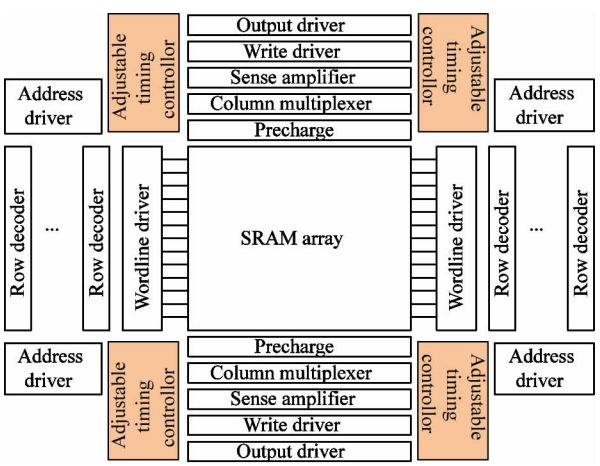


**图 1　多端口寄存器堆结构**

多端口寄存器堆的工作原理如下所示:时钟为低电平时,预充电模块把 SRAM Array 中的读位线预充为高电平;时钟上升沿到来时,根据读写使能信号开始读写操作。写操作时,写地址首先经过行地址译码器选通对应的行,将对应的字线置为高电平;外部写数据经过数据驱动电路转换成内部差分信号,经差分写位线把对应数据写入 SRAM Array 中已选通的行。当从 SRAM Array 读出数据时,对应读端口首先发出读地址,经译码后将对应的行字线拉高,SRAM Array 对应行中的数据首先输出到差分读位线,由经过灵敏放大器放大、锁存后输出,在读操作过程中,自适应时序控制模块将根据实际路径的延迟自动匹配读使能时序,在数据满足灵敏放大器的最小识别电压时开启使能信号 SAE,减小访存延迟,降低功耗,如图 2 所示。

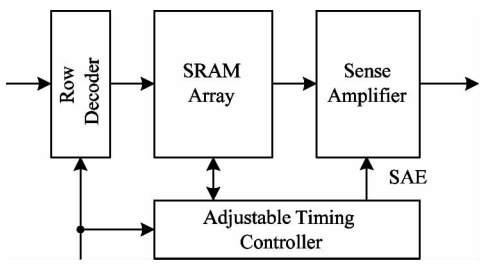


**图 2　自适应时序控制**

存储阵列是 SRAM 设计中重要的模块之一,理想的存储单元要具备面积小、存取电流大、读写裕量大、噪声容限大等特性。但这些特性很难同时满足,这就需要在面积、速度和功耗之间进行平衡。可用的存储单元有多种结构,除了经典的 6T CMOS 存储单元,还有 4T、4T2R、8T 等结构[10,11]。本文采用 6T 扩展结构,即有 2 个反相器交叉耦合构成的双稳态电路,采用差分信号进行读写操作;其理论静态功耗为零,具有较强的抗干扰能力;在读数模块增加两个 NMOS 以改进读取速率。

译码器是多端口寄存器的又一重要部件,所占芯片面积仅次于存储阵列[12]。译码器可以采用单级译码或多级译码结构来实现[13]。对于单级译码,一个 $n$ 位的译码器需要 $2n$ 个 $n$ 输入逻辑门,若输入引脚数超过 4 个,串联堆叠作用会产生较大的串联电阻和较长的延迟。为了优化译码电路的延时和功耗,本文采用多级译码结构,以 3 读 2 写 32 × 64 bit 寄存器堆为例,可采用两级静态译码:1 个预译码级(2-4 译码器)和 1 个最终译码级(3-8 译码器),其中预译码级产生 12 个中间信号作为最终译码器的输入,具体电路实现如图 3 所示。

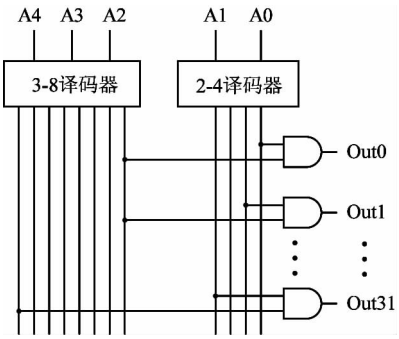


**图 3　二级译码器**

灵敏放大器用于在读操作中放大位线上毫伏级微小的电压差,加快读操作速度。在寄存器堆设计中,为了增加存储密度,存储单元通常使用最小尺寸或者略大于最小尺寸的晶体管。因此,读操作时位线电容通过存储单元放电的速度非常慢,尤其是当位线上连接的存储单元数量很多时,由于电容很大,造成读操作非常慢。通常采用灵敏放大器来放大位线上的电压差,加快读操作速度。一个快速、稳定、低失配和抗干扰强的灵敏放大器对于提高 SRAM 的读操作速度、降低读取失效率起关键作用。灵敏放大器有多种电路结构[14,15],本文中基于 Latch 结构来设计灵敏放大器,如图 4 所示,其具备速度快、工艺敏感度好等优点,在具体设计时可根据不同尺寸的寄存器堆进行 CMOS 管参数优化。

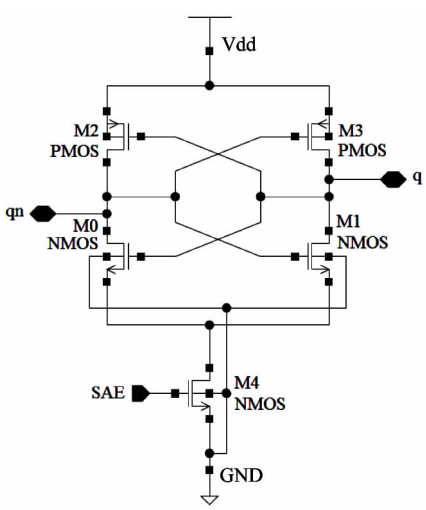


**图 4　灵敏放大器结构**

## 2　自适应反馈原理及实现

多端口寄存器堆访存延迟是寄存器堆设计的关键参数之一,直接影响寄存器堆的工作频率,访存延迟主要包括译码延迟和数据通路延迟:译码延迟指从有效地址输入到完成字线选通的延迟;数据通路延迟指从字线选通到寄存器堆输出有效信号之间的延迟,而其中的关键时序就是读操作时灵敏放大器打开和关闭的时刻,即灵敏放大器使能脉冲的控制。SRAM 中使用灵敏放大器检测并放大位线上的毫伏级的小摆幅电压,因此检测使能信号的时序非常重要。如图 5 所示,如果灵敏放大器使能脉冲 SAE 到来过早,位线上的摆幅(BL 与 BLB 差值)太小,锁存器可能由于噪声的作用而产生误翻转,进而使灵敏放大器识别出错误的数据;如果使能脉冲 SAE 太晚,位线上的摆幅积累过大,增加了不必要的延迟,又浪费了相当一部分的功耗。理想的状况是当位线差分输出信号差值恰好大于等于灵敏放大器的最小识别电压时,打开使能脉冲 SAE,这样既能保证时序又能减小读写延迟;而当灵敏放大器输出信号 $D_{out}$ 达到后续电路要求时,及时关闭位线、灵敏放大器和及其它输出模块。

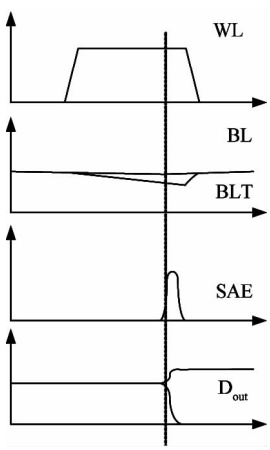


**图 5　寄存器堆访存时序**

在实际应用中,受到集成电路制造工艺偏差、电压和温度的影响,信号延迟容易波动,工艺和电压的波动影响寄存器堆电路各模块的时序,若灵敏放大器的使能信号不能很好地跟踪时序的变化,将会导致寄存器堆读写错误,因此,设计时需要在电路结构上对时序波动进行补偿。

传统方法采用 Replica 和 Chain Delay 的方法来跟踪时序的波动[16],如图 6 所示,其中图 6(a)采用不同的时钟相位来进行时序匹配,图 6(b)所示的

Chain Delay 方法采用固定延迟链来进行时序匹配。由于晶体管特征参数的波动随着晶体管特征尺寸的减小而不断加大,这类静态预补偿的方式难以精确地补偿实际芯片使用中的各种参数波动,实际效果较差。此外,为了保证时序匹配的准确度,这类补偿方法必须提供足够大的裕量,易造成性能与资源的浪费。

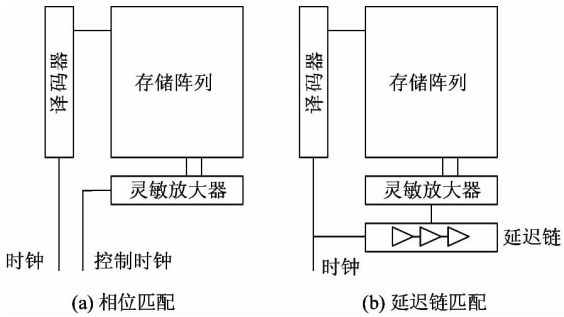


**图 6 传统的寄存器堆时序匹配原理**

本文基于自测试、可调节的延迟单元提出一种自适应延时补偿机制,它可以根据实际芯片工作需求来自动探知所需的补偿量,进而确定合理的时序,减少读延迟,降低功耗。本文中的多端口寄存器堆中使用自适应延迟模块来自动匹配灵敏放大器的使能控制信号,在位线上的电压差达到合适的摆幅后,延迟链的输出信号(也即灵敏放大器使能信号)变为高电平,灵敏放大器进入有效工作状态,进而将位线数据放大并输出。

自适应时序控制模块如图 7 所示,主要由参考序列寄存器(Ref __ Registers)、多路选择器(Mux)、比较器(Comparator)、可调延迟链(Adj _ Delay _ Chain)和自适应匹配状态机(Timing _ Auto _ Adjust _ FSM)等构成,其中 Ref _ Registers 存储的数据作为

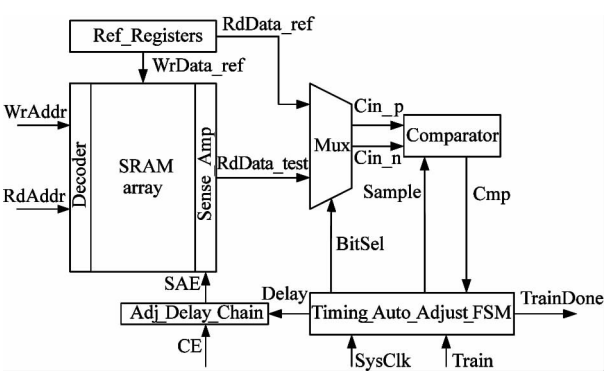


**图 7 自适应时序控制模块**

训练队列 WrData _ ref 写入寄存器堆,并提供比较器的参考端信号输入 RdData _ ref;由于存储阵列位数较多,为了节省硬件开销,采用循环扫描,每次循环通过 Mux 选择其中几位进入比较器进行对比,对比结果通过信号 Cmp 反馈给状态机;Timing _ Auto _ Adjust _ FSM 负责完成整个模块的状态控制,包括打开、关闭训练过程和根据比较结果调整延迟链大小;Adj _ Delay _ Chain 为动态可调延迟链,延迟控制信号 Delay 由 Timing _ Auto _ Adjust _ FSM 提供,延迟链输出信号作为灵敏放大器的使能控制信号 SAE。

当寄存器堆重新上电或者周边电路、温度等外部环境变化时,自适应时序控制模块开始工作,整个时序匹配过程主要包括下列步骤:

步骤 1:训练信号 Train 上升沿到来,系统进入自适应调整状态,首先将多端口寄存器堆中的延迟链初始化为典型工艺条件下延迟值,并由参考寄存器 Ref _ Registers 将训练序列写入寄存器堆阵列,对应地址记为训练地址。

步骤 2:在当前延迟链设置下对寄存器堆中训练地址进行访存操作,读出的训练数据经过 SRAM Array 中 Bitline、Column Multiplexer、Sense Amplifier 和 Output Driver 模块后记为 RdData _ Test。

步骤 3:读出序列 RdData _ Test 与参考序列 RdData _ Ref 经过 Mux 选择对应比特位分别在 Comparator 模块中进行比对,若两个序列的所有数据都一致,说明读写正确,此时延迟链裕量可能存在富余,则减小 Adj _ Delay _ Chain 中的延迟;若读出数据与参考值不一致,则增加 Adj _ Delay _ Chain 中的延迟。延迟链动态调整方式对匹配速度影响很大,本文采用了二分法来加速此过程。

步骤 4:执行完某次延迟链调整进程后,循环执行步骤 2 和步骤 3,直至找到能够正确读出数据的最小延迟,也即当前延迟链精度条件下最接近实际路径的延迟大小。此时自适应时序匹配进程完毕,将相关延迟链的配置信息写入配置寄存器。

自适应时序匹配的精度与延迟链最小调整幅度密切相关,最小调整幅度由延迟链的基本单元延迟决定,基本延迟单元越小,匹配精度越高;自适应时

序的匹配范围与延迟基本单元的数量密切相关,延迟单元个数越多,可匹配的范围越广。自适应匹配的匹配速度与延迟链调整方式密切相关,可以按照基本延迟单元进行逐次递增或递减,也可以按指数或者对数规律进行调整。

本文延迟链在以实际仿真和经验的基础上,覆盖寄存器堆访存路径在不同 PVT(工艺角、电压、温度的组合搭配)条件下的延迟波动范围,并留有一定裕量,同时为了减小硬件开销并加速搜索进程,本文采用二分法进行循环迭代。下面说明具体过程:

设定延迟链有 $n$ 个延迟单元,单个延迟单元大小为 $T_d$,则整个延迟链大小为 $nT_d$,令 $p = \log_2 n$,自适应时序匹配的具体过程如下:

(1) 第 1 次循环设定延迟为 $T_1 = \frac{1}{2}nT_d$。

(2) 若第 1 次读操作正确,则第 2 次循环设定延迟为 $T_2 = \left(\frac{1}{2} - \frac{1}{4}\right)nT_d$,否则第 2 次循环设定延迟为 $T_2 = \left(\frac{1}{2} + \frac{1}{4}\right)nt_d$。

(3) 若第 $p-1$ 次读操作正确,则第 $p$ 次循环设定延迟为 $T_p = T_{p-1} - T_d$,否则第 $p$ 次循环设定延迟为 $T_p = T_{p-1} + T_d$。

(4) 若第 $p$ 次读操作正确,则延迟链最终延迟为 $T = T_p$,否则延迟链最终延迟为 $T = T_p + T_d$。

在单次循环中,为了充分衡量读数据的正确性,可通过控制 Mux 电路来进行多次比对。

设定理想情况下,即位线电压差刚好等于灵敏放大器的最小识别电压时延迟为 $T_0$,经过上述 $p$ 个循环之后,此时延迟迭代得到的延迟 $T$ 与 $T_0$ 的差值小于 $T_d$,即 $0 \leq T - T_0 < T_d$。

下面以 $n = 8$ 为例阐述自适应匹配过程,延迟链长度为 8 个延迟单元,$p = \log_2 8 = 3$,也即通过 3 次循环迭代即可完成匹配进程。$Delay[2:0]$ 表示延迟链的设置情况,譬如当 $Delay[2:0] = 3'b101$ 时,延迟链长度为 5 个延迟单元 $5T_d$。

自动时序匹配的状态转移图如图 8 所示,假设访存路径实际延迟为 $4.3T_d$,自动匹配过程将分为三个循环从高位开始依次完成对 $Delay[2:0]$ 的配置,具体过程如下:

(1) 初次循环时令 $Delay[2:0] = 3'b100$,延迟链长度设置为中点,此时延迟链为 $4T_d$,由于延迟链长度小于实际延迟 $4.3T_d$,灵敏放大器开启时实际电压小于最小识别电压开启,经过 Comparator 多个周期信号比对,实际读出序列与参考序列不一致,Cmp 信号输出为 0,实际延迟不小于 $4T_d$,因此,$Delay[2] = 1'b1$。

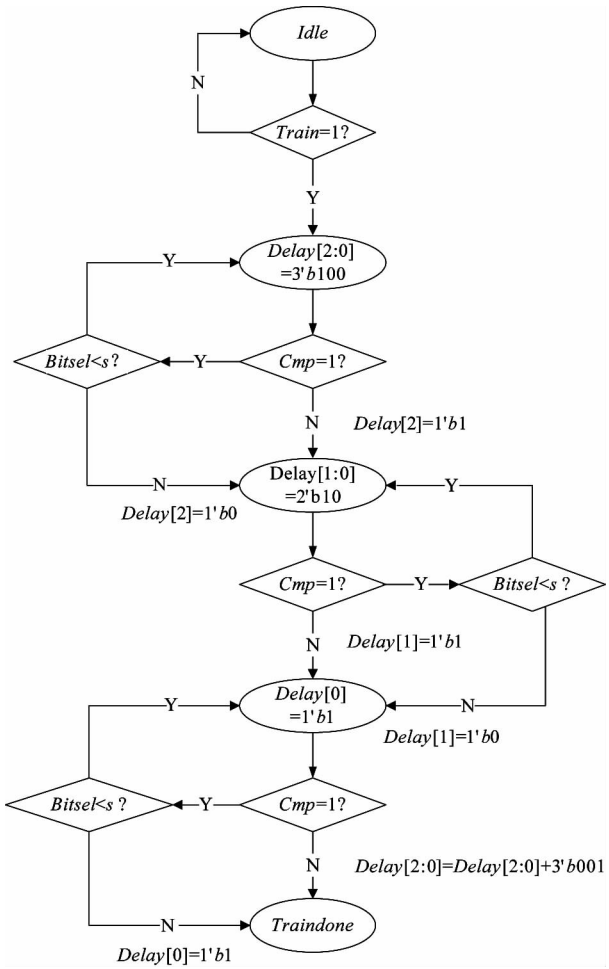


**图 8　自动时序匹配状态图($n=8$)**

(2) 第二次循环时首先令 $Delay[1:0] = 2'b10$,结合第一循环得到的 $Delay[2] = 1'b1$,此时 $Delay[2:0] = 3'b110$,延迟链长度为 $6T_d$,由于延迟链长度大于实际延迟 $4.3T_d$,灵敏放大器开启时实际电压大于最小识别电压开启,经 $s$ 个周期完成序列比对后,实际读出序列与参考序列一致,Cmp 始终为 1,因此,$Delay[1] = 1'b1$。

(3) 第三次循环时首先令 $Delay[0] = 1'b1$,结合前面两次循环得到的 $Delay[2] = 1'b1$ 和

$Delay[1] = 1'b0$，此时 $Delay[2:0] = 3'101$，此时延迟链长度为 $5T_d$，由于延迟链长度大于实际延迟 $4.3T_d$，灵敏放大器开启时实际电压大于最小识别电压开启，经 $s$ 个周期完成序列比对后，实际读出序列与参考序列一致，Cmp 始终为 1，因此 $Delay[0] = 1'b1$。

完成三次循环后，$Delay[2:0] = 3'b101$，也即延迟链长度最终设置为 $5T_d$，考虑到延迟链可能的配置依次为 $T_d$，$2T_d$，…，$8T_d$，实际延迟为 $4.3T_d$，因此自适应匹配方法可求解出最为匹配的时序延迟 $5T_d$。

Questasim 软件对此次循环的仿真波形如图 9 所示，$qvDlyAdj[2:0]$（状态机中 $Delay[2:0]$）为延迟链控制信号，展示了延迟链变化轨迹和循环迭代过程，第一个循环为 100，第二个循环为 110，然后迭代到 101，仿真结果符合自适应匹配方法的状态转移过程，得到最为匹配的时序延迟。

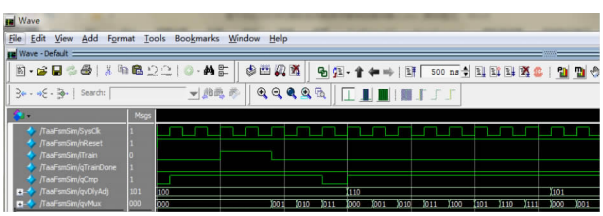


**图 9　状态机仿真图**

为了增加匹配精度，我们可以采用 16 个延迟单元为 $0.5T_d$ 的延迟链，此时延迟链控制位为 $Delay[3:0]$，延迟链可能的配置为 $0.5T_d$，$1.0T_d$，$1.5T_d$，…，$7.5T_d$，$8.0T_d$，经过四个循环迭代后，自适应匹配得到的延迟为 $4.5T_d$。

自适应的时序匹配过程需要循环迭代，很难完成实时精确匹配，为了减小外界变化对寄存器堆时序稳定度的影响，本文在自适应匹配的基础上额外增加一个延迟裕量，在保证延迟性能的基础上提高了访存稳定度。裕量大小与寄存器堆具体工作环境有关，可通过对不同条件下 PVT 的模拟仿真获取延迟波动范围并设置合适的延迟裕量大小。

## 3　实验结果

本文基于 SMIC 40nm 工艺，采用上文阐述的结构设计完成 3 读 2 写 32 × 64 bit 寄存器堆，具体参数如表 1 所示。

**表 1　寄存器堆参数**

| Process | SMIC 40nm |
|---|---|
| Depth(Words) | 32 |
| Width(Bits) | 64 |
| Column Mux | 1 |
| Frequency | ≥1GHz |
| Read ports | 3 |
| Write ports | 2 |
| Activity factor | 50 |
| Area | 8821.05μm$^2$ |
| Access time | 357ps |

为了更精确地评估寄存器堆性能和自适应模块的面积开销，基于 Cadence 公司的 Virtuoso 平台完成了版图设计（图 10），工艺库采用 SMIC 40nm，3 读 2 写 32 × 64 bit 寄存器堆面积为8821.05 μm$^2$，其中自适应匹配和调优模块的面积89.8μm$^2$，约占总面积的 1%，自适应模块额外开销极小。

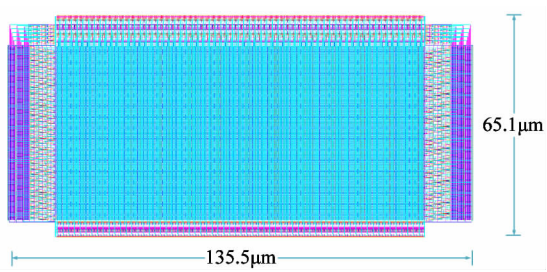


**图 10　寄存器堆（3 读 2 写 32 × 64bit）版图**

寄存器堆在 TT Corner、Input Transition = 10ps 和 Output Capacitance = 0.025ps 时的读写时序如图 11所示，SysClk 为系统时钟信号，WrEn 和 WrData 信号为写使能信号和写入数据，RdEn 和 RdData 信号为读使能信号和读出数据。由波形可见，在 SysClk 的第 3 个和第 6 个时钟上升沿，分别将数据“1”和“0”写入寄存器堆，由于读使能一直为高，在每个时钟周期寄存器堆都进行数据读取，数据输出端默认值为“0”；在第3个时钟周期写入“1”之后，寄存器堆输出端变为“1”并一直保持，直到第 6 个时钟周期写入“0”之后，输出端也变为“0”。

为了全面衡量寄存器堆的自适应时序匹配和调优效果，以传统的ChainDelay方法作为参照，在不

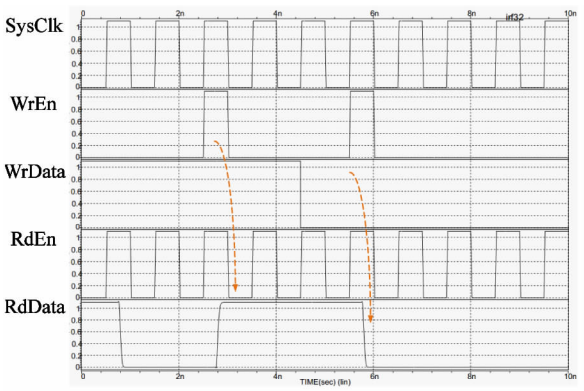


**图 11　3 读 2 写 32 ×64bit 寄存器堆读写波形图**

同电源电压、温度和工艺角情况下,全面比较了自适应方法和 Chain Delay 方法在延迟和功耗上的性能,其中自适应模块的基本调谐延迟单元 $T_d$ 为 25ps。

由表 2 可见,采用自适应匹配和调优后,多端口寄存器堆的延迟得到改善,延迟减小约 22%。这是由于传统的预设匹配的方法为了保证读数据的稳定性,时序裕量比较大;而采用自适应方法后,可根据外界环境进行自适应匹配和优化,将灵敏放大器的实际使能控制时间与理想值的差距控制在 1 个延迟调谐单元之内。

**表 2　自适应方法与传统 Chain Delay 方法读延迟时间比较**

| | | TT Corner | | | FF Corner | | | SS Corner | | |
|---|---|---|---|---|---|---|---|---|---|---|
| *T*(℃) | *Vd*(V) | *Cd* | *Taa* | *Imp* | *Cd* | *Taa* | *Imp* | *Cd* | *Taa* | *Imp* |
| -40 | 0.99 | 498 | 434 | 13% | 421 | 323 | 23% | 689 | 533 | 23% |
| 25 | 0.99 | 519 | 418 | 19% | 433 | 354 | 18% | 705 | 548 | 22% |
| 125 | 0.99 | 547 | 439 | 20% | 448 | 353 | 21% | 723 | 575 | 20% |
| -40 | 1.1 | 415 | 321 | 23% | 369 | 266 | 28% | 486 | 413 | 15% |
| 25 | 1.1 | 430 | 341 | 21% | 382 | 282 | 26% | 515 | 450 | 13% |
| 125 | 1.1 | 450 | 368 | 18% | 398 | 304 | 24% | 558 | 448 | 20% |
| -40 | 1.21 | 369 | 276 | 25% | 336 | 234 | 30% | 415 | 329 | 21% |
| 25 | 1.21 | 384 | 295 | 23% | 349 | 249 | 29% | 434 | 352 | 19% |
| 125 | 1.21 | 404 | 318 | 21% | 366 | 270 | 26% | 459 | 382 | 17% |
| Average | | 446 | 357 | 20% | 389 | 293 | 25% | 554 | 448 | 19% |

**说明**:延迟时间单位为 ps;T 为温度,单位为℃; $V_d$ 为电源电压,单位为 V; $C_d$ 表示传统的 Chain Delay 时序匹配方法, *Taa* 表示自适应时序匹配方法; *Imp* 表示 *Taa* 相对于 $C_d$ 的优化百分比;Average 表示算术平均值。

**表 3　自适应方法与 Chain Delay 方法读写功耗比较**

| | | TT Corner | | | FF Corner | | | SS Corner | | |
|---|---|---|---|---|---|---|---|---|---|---|
| *T*(℃) | *Vd*(V) | *Cd* | *Taa* | *Imp* | *Cd* | *Taa* | *Imp* | *Cd* | *Taa* | *Imp* |
| -40 | 0.99 | 1.19 | 0.75 | 37% | 1.36 | 0.76 | 45% | 1.12 | 0.62 | 45% |
| 25 | 0.99 | 1.24 | 0.7 | 44% | 1.13 | 0.73 | 36% | 1.21 | 0.69 | 43% |
| 125 | 0.99 | 1.01 | 0.73 | 28% | 1.12 | 0.81 | 28% | 1.29 | 0.69 | 47% |
| -40 | 1.1 | 1.66 | 1.4 | 16% | 1.9 | 1.06 | 44% | 1.42 | 0.85 | 40% |
| 25 | 1.1 | 1.32 | 0.88 | 33% | 1.46 | 0.92 | 37% | 1.16 | 1.02 | 12% |
| 125 | 1.1 | 1.37 | 0.91 | 34% | 1.45 | 1.01 | 30% | 1.43 | 0.87 | 39% |
| -40 | 1.21 | 2.31 | 1.37 | 41% | 2.41 | 1.64 | 32% | 1.89 | 1.47 | 22% |
| 25 | 1.21 | 1.68 | 1.06 | 37% | 1.83 | 1.1 | 40% | 1.53 | 1.00 | 34% |
| 125 | 1.21 | 1.73 | 1.14 | 34% | 2.19 | 1.29 | 41% | 1.55 | 1.08 | 31% |
| Average | | 1.50 | 0.99 | 34% | 1.65 | 1.04 | 37% | 1.40 | 0.92 | 0.35 |

**说明**:功耗单位为 mW;T 为温度,单位为℃; $V_d$ 为电源电压,单位 V; $C_d$ 表示传统的 Chain Delay 时序匹配方法, *Taa* 表示自适应时序匹配方法; *Imp* 表示 *Taa* 相对于 $C_d$ 的优化百分比;Average 表示算术平均值。

3 读 2 写 32 ×64 bit 寄存器堆的单端口读写功耗见表3,相比于传统的 Chain Delay 方法,基于自适应时序匹配的寄存器堆功耗减小约35% 。这是因为自适应匹配可有效减少寄存器堆中耗电部件的开启时间,在完成访存操作后及时关闭相关电路模块,降低系统功耗。

## 4 结 论

多端口寄存器堆模块是现代低功耗高性能处理器与 SoC 的重要部件之一,直接影响芯片性能与稳定性。在设计中对关键路径延时进行预匹配、预留 PVT 冗余量的传统方法,已不足以高效地应对纳米时代工艺下固有的参数随机波动加大现象。通过本文提出的可控的、自测试、自适应关键路径延时匹配与自动调优算法与电路,可以有效跟踪上述随机变化、自动进行调优匹配与补偿,以更灵活有效的方式剔除设计中的过多冗余量,在保证可靠性与稳定性的前提下,提高电路的性能、降低功耗,同时改善成品率、降低成本。本文基于 SMIC 40nm 工艺下完成 3 读 2 写 32 ×64bit 多端口寄存器堆的电路和版图设计,实验结果显示:自适应匹配模块面积开销小,仅占寄存器堆总面积的 1% ;对寄存器堆访存性能提升明显,访存延迟仅为 357ps;同传统的固定延迟链构成的 Chain Delay 技术相比,延迟改进 22% ,功耗减小 35% 。

**参考文献**


[1] Zheng N, Mazumder P. Modeling and mitigation of static noise margin variation in subthreshold sram cells[J]. *IEEE Transactions on Circuits & Systems I: Regular Papers*, 2017, 64(10):2726-2736

[2] Mittal S. A survey of techniques for designing and managing CPU register file[J]. *IEEE Transactions on Parallel and Distributed Systems*, 2017, 28(1):16-28

[3] Sarfraz K, Chan M. A 1. 2V-to-0. 4V 3. 2GHz-to-14. 3MHz power-efficient 3-port register file in 65-nm CMOS[J]. *IEEE Transactions on Circuits & Systems I: Regular Papers*, 2017, 64(2):360-372

[4] Agbo I, Taouil M, Kraak D, et al. Integral impact of BTI, PVT variation, and workload on SRAM sense amplifier[J]. *IEEE Transactions on Very Large Scale Integration Systems*, 2017, 25(4):1444-1454

[5] Amrutur B S, Horowitz M A. A replica technique for wordline and sense control in low-power SRAM's[J]. *IEEE Journal of Solid-State Circuits*, 1998, 33(8):1208-1219

[6] Song Y, Yu H, Dinakarrao S M P. Reachability-based robustness verification and optimization of SRAM dynamic stability under process variations[J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2014, 33(4):585-598

[7] Gong N, Wang J, Sridhar R. Variation aware sleep vector selection in dual Vt dynamic or circuits for low leakage register file design[J]. *IEEE Transactions on Circuits & Systems I: Regular Papers*, 2017, 61(7):1970-1983

[8] Singh J, Mohanty S P, Pradhan D. Robust SRAM Designs and Analysis[M]. New York: Springer, 2013. 31-35

[9] Jinhui W, Lina W, Haibin Y, et al. cNV SRAM: CMOS technology compatible non-volatile SRAM based ultra-low leakage energy hybrid memory system[J]. *IEEE Transactions on Computers*, 2016, 65(4):1055-1067

[10] Yasuhiro M, Hidehiro F, Hiroki N, et al. An area-conscious low-voltage-oriented 8T-SRAM design under DVS environment[C]. In: Proceedings of the IEEE Symposium on VLSI Circuits, Kyoto, Japan, 2007. 256-257

[11] Zhang H, Chen X, Xiao N, et al. Architecting energy-efficient STT-RAM based register file on GPGPUs via delta compression[C]. In: Proceedings of the ACM/EDAC/IEEE Design Automation Conference (DAC), Austin, USA, 2016. 1-6

[12] Mishra A K, Acharya D P, Patra P K. Novel design technique of address decoder for SRAM[C]. In: Proceedings of the 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies, San Francisco, USA, 2014. 1032-1035

[13] Pavlov A, Sachdev M. CMOS SRAM Circuit Design and Parametric Test in Nano-Scaled Technologies[M]. New York: Springer, 2010. 31-33

[14] Rodrigues S, Bhat M S. Impact of process variation induced transistor mismatch on sense amplifier performance [C]. In: Proceedings of the International Conference on

Advanced Computing and Communications, Surathkal, India, 2006. 497-502

[15] Licciardo G D, Cappetta C, Benedetto L D, et al. Design of an offset-tolerant voltage sense amplifier bit-line sensing circuit for SRAM memories[J]. *Electronics Letters*, 2016, 52(16):1372-1373

[16] Arandilla C D C, Madamba J A R. Comparison of replica bitline technique and chain delay technique as read timing control for low-power asynchronous sram[C]. In: Proceedings of the 5th Asia Modelling Symposium, Manila, Philippines, 2011. 275-278

# A low latency register file based on adjustable access latency

Yuan Guojun[*,**], Shen Hua[*], Shao En[*,**], Zang Dawei[*]

([*]Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)

([**]Institute of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing 100049)

**Abstract**

It is pointed that the random variation of the characteristic parameters of semiconductor process and transistors gets bigger with the decrease of the chip feature size, thus the traditional register file design based on prematch has to increase the matching margin to ensure the reliability of read and write operations. To overcome this key factor of restricting register file performance, a low power register file circuit structure based on adjustable access latency is proposed. The proposed mechanism can auto-test the practical path delay of the sense amplifier, and automatically match and tune time delay of sense enable signals to guarantee the correct operation, so as to improve the performance and power of the circuit by reducing unnecessary margin pre-placed in design. For 3-read ports and 2-write ports 32×64bit register file generated in SMIC 40nm technology, its area is 135.5μm×65.1μm and read access latency is 357ps. The simulation results show that compared with the traditional chain delay technique, the read access latency and the power consumption of the mechanism are reduced by 22% and 35% respectively.

**Key words**: multi-port register file, adjustable access latency, low latency, low power, static random access memory (SRAM)