

一种 32 位异步嵌入式微处理器原型

王 蕾, 邓 宇, 王志英, 戴 葵

(国防科技大学 计算机学院, 湖南 长沙 410073)

摘 要:异步集成电路设计技术很好地解决了深亚微米工艺条件下同步集成电路设计技术面临的问题。文中在对一系列关键技术进行研究的基础上,设计并实现了一款 32 位异步嵌入式微处理器原型。在基于宏单元异步集成电路设计流程的基础上,结合解同步技术,提出了异步嵌入式微处理器原型的设计流程。研究了如何实现异步嵌入式微处理器的精确异常、相关检测、同步异步接口和本地握手电路等。最后给出了原型的实现和初步的性能评测结果。

关键词:异步集成电路;嵌入式微处理器;设计流程

中图分类号:TP368.1

文献标识码:A

文章编号:1673-629X(2008)01-0202-04

A 32b Asynchronous Embedded Microprocessor Prototype

WANG Lei, DENG Yu, WANG Zhi-ying, DAI Kui

(School of Computer, National University of Defense Technology, Changsha 410073, China)

Abstract: In this paper, designed a prototype of 32b asynchronous embedded microprocessor based on architecture specification of a 32b microprocessor architecture "Tengyue-1". The design methodology is based on the combination of macro cell design and de-synchronization techniques. The issues about microarchitecture design are discussed, such as implementing precise interruption, dependence determination, interface between asynchronous and synchronous circuits and local handshake circuits. At the end of the paper, the preliminary results of performance evaluation were given.

Key words: asynchronous circuits; embedded microprocessor; design flow

1 简介

在深亚微米的工艺条件下,单芯片集成的晶体管数越来越多,芯片面积越来越大,很难使用全局时钟同步(Synchronize)芯片各个部分的操作,因而不可避免地出现了时钟扭曲难以解决、时钟功耗过大等问题。随着嵌入式微处理器广泛的应用于手机等设备,对嵌入式微处理器的性能、功耗和电磁兼容性等指标的要求越来越高。异步集成电路没有全局时钟,具有低功耗和高电磁兼容性等特点,再次成为学术界的研究热点^[1,2]。

同步电路是时钟驱动的电路上,可以认为所有的信号都只在时钟的上升沿或者下降沿有效。异步电路是事件驱动的电路上,各模块通过握手信号进行交互,保证操作的顺序,实现模块间的同步和通信。英国 Manchester 大学由 Steve Furber 教授领导的先进处理器技术

研究小组从 20 世纪 90 年代初期就开始了异步电路设计思想的研究^[3,4]。他们的工作表明异步集成电路是很有潜力的,在性能和功耗上完全可以超越相同工艺的同步集成电路。文中对一款 32 位异步微处理器原型的设计进行了研究。

2 设计流程

在先前的工作中^[5],已经设计和实现了同步嵌入式微处理器芯片“腾越-1”。在此基础上,将同步微处理器的指令流水线单独提取出来,作为实现异步微处理器原型的基础。为了叙述方便,从下文开始称同步流水线为 SPC(Synchronous Pipeline Core),称对应的异步流水线,也就是异步微处理器原型为 APC(Asynchronous Pipeline Core)。

异步嵌入式微处理器的设计和实现采用解同步技术和基于宏单元的异步集成电路设计流程^[6]相结合的设计方法。设计流程如下所示:

- 1) 从同步微处理器“腾越-1”的 RTL 级源代码中将同步流水线 SPC 的源代码抽取出来;
- 2) 将同步流水线的 RTL 描述转换为基于锁存器

收稿日期:2007-03-09

基金项目:国家自然科学基金资助项目(90407022)

作者简介:王 蕾(1977-),女,陕西西安人,博士,研究方向为异步集成电路、微处理器设计、高性能处理器体系结构技术。

注:中国计算机学会微机(嵌入式)专委会学术会议优秀论文。

的同步电路;

3)使用 STG 描述本地握手电路的协议,使用 Pet-rify^[7]进行综合,得到本地握手电路的网表;

4)采用同步电路基于锁存器的重定时技术^[8,9],对数据通路进行重定时,改善电路的性能。

5)数据通路采用同步电路的综合工具(如 Design Compiler)综合得到网表,和本地握手电路网络的网表连接起来,得到解同步的异步流水线 APC 的网表。

6)应用基于宏单元的异步集成电路设计流程,将关键单元全定制为宏单元,交付后端设计。

3 体系结构设计

异步微处理器原型 APC 的体系结构如图 1 所示。指令流水线分为 5 段,分别是取指段 IF、译码段 ID、执行段 EX、访存段 MEM 和写回段 WB。APC 各流水段的功能和 SPC 相同。图 1 中,灰色的矩形表示主锁存器和从锁存器,本地握手控制电路互相连接生成锁存器的控制信号。

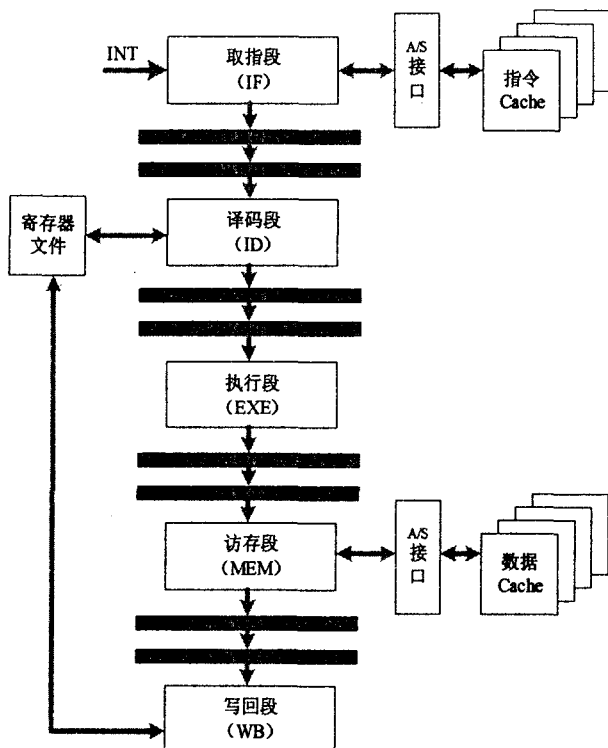


图 1 APC 的体系结构

接下来讨论异步嵌入式微处理器原型 APC 体系结构设计过程中的关键问题。

3.1 精确异常

异步流水线采用握手信号来表示计算的完成,没有确定的相对于某个全局信号(例如时钟)的时刻来观测电路的状态,因而实现精确异常难度很大。

APC 为单发射体系结构,其指令的结束顺序和发

射顺序是相同的。按照精确异常的定义,异常处理机制必须保证,引起异常的指令和在流水线中位于该指令后、异常处理程序第一条指令前的所有指令都不会修改计算的全局状态,这些指令被称为“取消的指令”。

采用作废这些指令的方法来保证异常处理的精确。当流水线检测到异常发生时,通知 MEM 段和 WB 段。除了访存指令,所有的指令都是执行到 WB 段才修改系统状态,因此可以在 WB 段作废“取消的指令”。对于访存指令,在 MEM 段作废。同时,IF 段从异常处理例程开始取指,并设置指令标记位为有效,当 WB 段执行的指令为有效时,则停止作废指令。

3.2 相关检测

流水线处理器都存在相关问题,包括数据相关、控制相关和结构相关。由于 APC 是单发射处理器,因而不存在结构相关,这里就不做讨论。

数据相关:在异步微处理器原型 APC 中,数据相关的判定通过一个 3 段的先进先出队列 FIFO 来实现。该 FIFO 中保存前 3 条在 ID 段译码的指令的目的操作数的地址。当指令在 ID 段译码的时候,如果当前指令的源操作数的地址和 FIFO 中任何一项的目的操作数地址相匹配,则检测到了数据相关。APC 通过在流水线的 EX 段插入空操作(NOP)来消除相关。

控制相关:分支指令和异常都会引起控制相关。异步流水线 APC 使用两条空操作指令填充分支延迟槽,当 EX 段判断分支跳转成功,则反馈给 IF 段新的目标地址,IF 段从新地址开始取指。

4 电路设计

4.1 同步异步接口

由于 APC 的指令 Cache 控制器和数据 Cache 控制器都采用原有的“腾越-1”的同步模块,因而这两个接口都是同步电路和异步电路通信的接口。同步子系统和异步子系统通信的关键问题是防止同步失败(Synchronization Failure)问题。因而如果一个系统既包括同步系统,也包括异步系统,必须使用同步机制来防止同步失败问题的出现。在设计 APC 的指令 Cache 和数据 Cache 接口时,文中借鉴 GALS(Global Asynchronous Locally Synchronous)技术,采用^[10]中用于 GALS 点到点互连的通信机制,来实现同步子系统和异步子系统的通信。按照通信方向的不同,设计分为异步到同步接口和同步到异步接口,下面将分别讨论。

(1)异步到同步接口:图 2 给出了异步到同步接口的示意图。当没有数据的时候,输出时钟 clkB 被取反,输入仲裁器 Arbiter,这时 Arbiter 选择同步电路。接口电路采用两段握手协议。异步生产者产生新的数

据由 req 信号指示,当时钟 clkB 为高时,Arbiter 选择异步电路。如果时钟的下降沿和 req 信号的变化几乎同时,则会出现亚稳态问题。但是电路中的 Arbiter 使得在 req 信号传播的时候,时钟信号是停止的,因而避免了亚稳态问题的出现。当 Arbiter 选择异步电路的时候,数据被锁存,然后释放 Arbiter。这时同步时钟启动,同步电路将数据锁存到最后一级锁存器中,req 信号同时也被锁存,并将输出反馈给异步电路作为应答信号 ack。

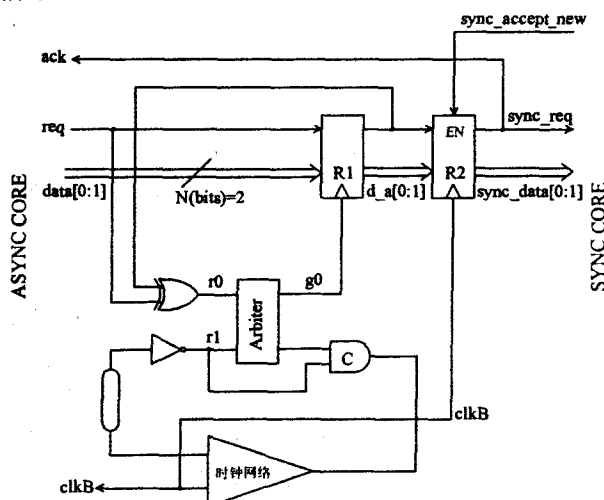


图 2 异步到同步接口

(2)同步到异步接口:图 3 给出了同步到异步接口的示意图,和异步到同步接口基本相同。设计的难点在于知道何时异步电路能够接收数据,所以主要的困难在于如何将控制信号从异步电路传递到同步电路。唯一的区别在于将数据从同步电路传递到异步电路的方式,遵循两段握手协议的请求信号和数据打包一起发送,通知异步电路新的数据到达,由于同步状态机总是等待 sync_ack 信号有效才发送新的数据,req 信号能够变化,保证数据传输的正确。

4.2 本地握手电路

图 4 给出了相邻锁存器的控制电路框图,A 和 B 为锁存器的控制信号,图 5 给出了相邻的 EVEN 锁存器 A 和 ODD 锁存器 B 的 SDFLC 的 STG。

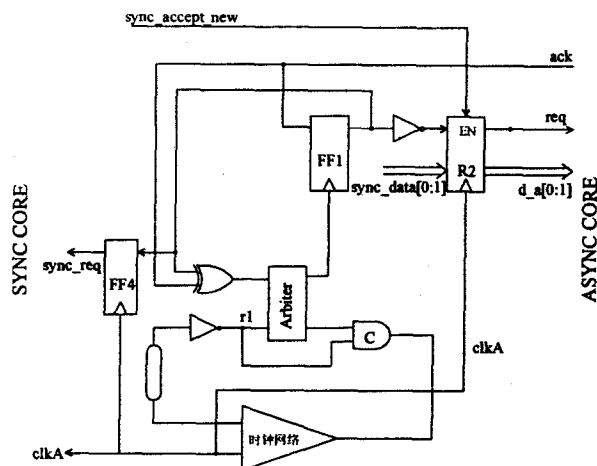


图 3 同步到异步接口

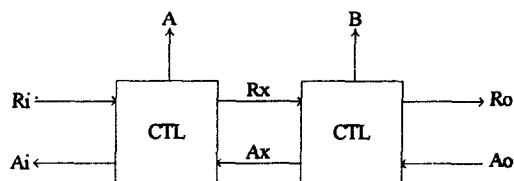


图 4 相邻的锁存器控制器框图

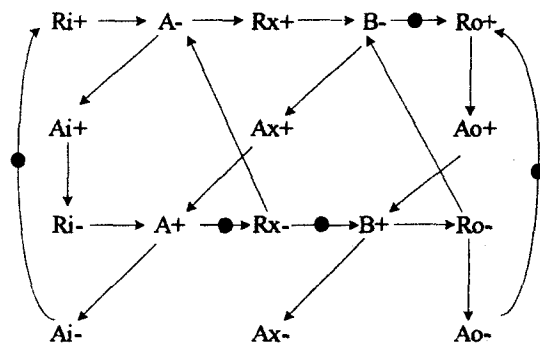


图 5 锁存器控制器的 STG

使用 Petrifly 综合一组握手协议(控制一对 EVEN 锁存器和 ODD 锁存器),得到了一对锁存器的本地握手控制电路,再将若干个本地握手电路按照解同步网表所确定的拓扑关系连接起来,就得到了最终的异步控制电路。图 6 给出了 EVEN 锁存器和 ODD 锁存器的本地握手控制电路。

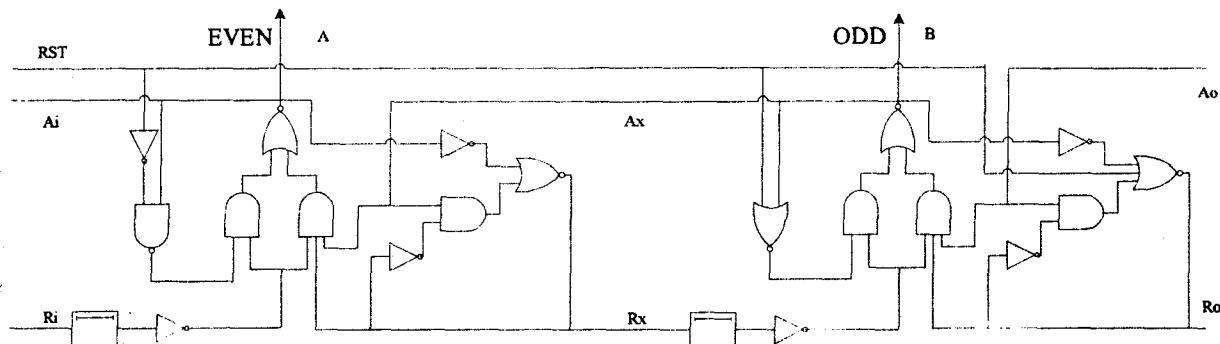


图 6 本地握手电路

5 实 现

假设主锁存器由同步时钟的上升沿控制,从锁存器由同步时钟的下降沿控制。在 Modelsim Designer 中模拟数据通路,使用测试 SPC 所使用的测试程序集,从而保证在替换锁存器的过程没有引入错误。通过测试程序集的测试后,数据通路使用 Design Compiler 综合,约束各个锁存器组之间的最大延迟。得到的延迟信息作为匹配延迟单元的延迟参考值。得到网表后,再使用同样的测试集,用同步时钟对网表的正确性进行测试。测试在 Modelsim Designer 中进行,将综合得到的延迟信息 SDF 文件反标给网表,进行前仿。在设计控制通路时,使用 Petrify 综合一组控制器的 STG 描述,按照同步网表的拓扑关系组合,得到整个电路的本地握手控制电路。控制通路的测试通过模拟输入端的请求序列观测输出的锁存器控制信号是否正确。

对数据通路和控制通路的测试正确之后,通过手工组合数据通路和控制通路,得到异步微处理器原型 APC 的网表。先使用模拟延迟单元代替真正的延迟单元对整个电路进行整合和测试。这时再次使用“腾越-1”的测试程序集。测试时,系统仅包括异步微处理器原型 APC、指令 Cache 和数据 Cache。在测试开始时,程序已经加载在指令 Cache 存储器中。

6 性能评测

将异步微处理器的原型系统 APC 的网表和同步微处理器的指令流水线核 SPC 的反标了 SDF 的网表在 Modelsim Designer 中进行模拟,使用相同的测试程序,比较异步指令流水线和同步指令流水线的性能,使用 9 个 ARM 基准测试程序进行测试。同步流水线 SPC 的工作主频为 266MHz,异步微处理器原型 APC 以其固有的速率运行。图 7 给出了 9 个测试程序在异步微处理器原型上的执行时间减去在同步微处理器上的执行时间得到的结果。可以看到,在绝大多数情况下,异步微处理器原型的性能要优于同步微处理器的性能。

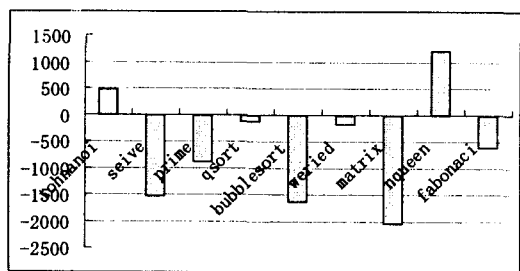


图 7 程序执行时间比较(异步—同步)

对 APC 和 SPC,以及将 SPC 的触发器 FF 替换为

主从锁存器所得到的基于锁存器的同步电路在 Design Compiler 中进行综合,得到了三者的面积,并进行了比较,如表 1 所示。综合所采用的工艺为 0.18 μ m 工艺。

表 1 面积比较

	面积(mm ²)
基于触发器的同步网表	0.47
基于锁存器的同步网表	0.41
异步网表	0.45

从表 1 可见,基于锁存器的同步网表的面积要小于基于触发器的同步网表,这是因为流水线电路中使用大量的存储器件,而锁存器比触发器的面积要小,因此将触发器替换为主从锁存器能够减少电路的面积。还可以看到,解同步网表的面积略大于基于锁存器的同步网表,这是由于异步控制通路面积的耗费。

7 结 论

研究了异步嵌入式微处理器原型 APC 设计和实现的关键技术,最后给出了对原型进行性能评测的初步结果。进一步的工作将完善该原型的设计,对原型进行分析和优化,最终实现异步嵌入式微处理器芯片。

参考文献:

- [1] Peeters A. The 'asynchronous' bibliography(BIBTEX)database file async. bib[EB/OL]. 2006. <ftp://ftp.win.tue.nl/pub/tex/async.bib>. Z, corresponding e-mail address: async-bib@win.tue.nl.
- [2] Garside J. The Asynchronous Logic Homepage[EB/OL]. 2006. WWW:<http://www.cs.man.ac.uk/amulet/async/>.
- [3] Furber S B, Day P, Garside J D, et al. AMULET1: A micropipelined ARM[C]//Proceedings IEEE Computer Conference (COMPCON). San Francisco, CA, USA: IEEE Computer Press, 1994:476-485.
- [4] Furber S B, Garside J D, Temple S, et al. AMULET2e: An Asynchronous Embedded Controller[C]//Proceedings of 3rd International Symposium on Asynchronous Circuits and Systems. Eindhoven, The Netherlands: IEEE Computer Press, 1997:290-299.
- [5] 王 蕾,陆洪毅,王 进,等.一种面向嵌入式应用的片上系统:腾跃-1[J].电子学报,2005,33(11):2036-2039.
- [6] 龚 锐,王 蕾,戴 葵,等.异步集成电路 C 标准单元的设计和实现[J].微电子学与计算机,2005,22(1):144-147.
- [7] Cortadella J, Kishinevsky M, Kondratyev A, et al. Petrify: a tool for manipulating concurrent specifications and synthesis of asynchronous controllers[J]. IEICE Transactions on Information and Systems, 1997,80:315-325.
- [8] Maheshwari N, Sapatnekar S. Optimizing large multiphase lev-

(下转第 208 页)

.....

读取实时数据,主要通过语句 URL u = new URL (“http://192.168.0.9/TempData.txt”)锁定嵌入式 Web 服务器在网络中的位置,其中,192.168.0.9 为 Web 服务器的 IP 地址,Thread.sleep(3000)语句用于设置采集的时间间隔,时间间隔为 3s,采集时间可以根据现场采集时间的需要更改采集参数。

2)在 PC 机上安装了 Java 的运行环境,实验中安装的是 jsk-6-windows 版本。在编译时,进入 MonitorApplet.java 文件的存放路径,输入语句:javac MonitorApplet.java 可编译 MonitorApplet.java 文件,生成一文件 MonitorApplet.class。

3)在设计用于采集的 HTML 网页中加入代码:<applet code = “MonitorApplet.class” name = “MonitorApplet” width = 400 height = 200 ></applet>,就完成了将 Applet 嵌入到了 HTML 页面中,网页部分代码:

```
<html>
<head></head>
<title>监视页面</title>
<body>
  <a href = “index.html”>首页</a>
  <a href = “webMonitor.html”>传感器监测</a>
  .....
  <applet code = “MonitorApplet.class” name = “MonitorApplet”
width = 400 height = 200 >
  </applet>
</body>
</html>
```

最后,在已加载了 ucos-II 和 Uip 程序的嵌入式 Web 服务器上添加嵌入了 Applet 的 webMonitor.html 网页,同时,将 MonitorApplet.class 文件转换成十六进制代码,置于 Uip 的 fsdata.c 文件中嵌入到处理器的 Flash 里,通过 Keil 将所有的代码编译生成 Hex 文件,下载到嵌入式 Web 上。

此时,用户在客户端就可以通过浏览器输入 IP 地址 192.168.0.9(一定要与嵌入在 Web 服务器的 IP 一

致)访问在嵌入式 Web 服务器上采集的实时数据。实验结果显示如图 5 所示。

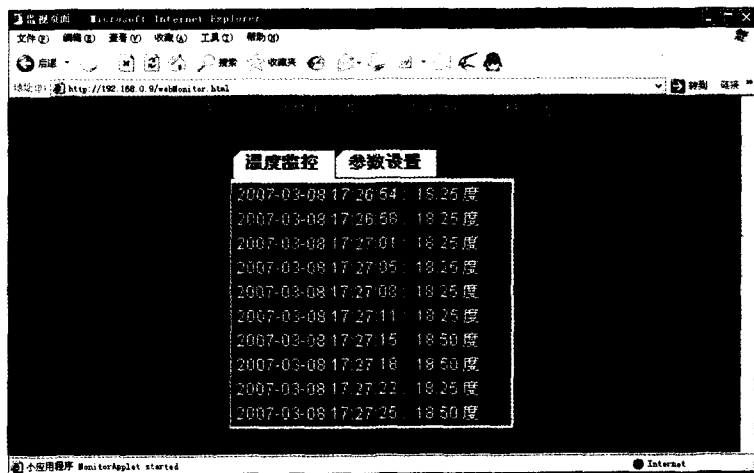


图 5 在客户端的实时数据的显示网页

3 总 结

介绍了 Web 服务器的工作原理及关键技术。基于 P89V51RD2 平台和 ucos-II 操作系统,将 CGI 和 Java Applet 技术内嵌于嵌入式 Web 服务器上,通过客户端的浏览器完成对远程监控系统数据的实时采集,节约了大量的硬件资源和人力。该采集技术具有通用性,适用于任何嵌入式 Web 服务器,应用范围相当广,不仅仅是远程,还能满足区域、工厂和车间等不同情况下测控的需要。

参考文献:

- [1] 韩光洁,赵海,王金东,等. Embedded Internet 技术及其综述[J]. 小型微型计算机系统, 2004, 25(5): 798-802.
- [2] He Pengju, Dai Guanzhong, Fu Lei, et al. Distributed Data Collection and Supervision Based on Web Sensor[C]//Proceedings of SPIE. [s.l.]: [s.n.], 2006.
- [3] 邓治国,张维新. uIP TCP/IP 协议栈在 51 系列单片机上的应用[J]. 微计算机信息, 2004, 20(3): 88-90.
- [4] 张冬宇,郑崇苏. 实时操作系统下单片机嵌入 TCP/IP 的研究[J]. 福州大学学报: 自然科学版, 2006, 34(4): 224-228.
- [5] 郭颂,杨开英,李朝纯. Java Applet 程序网络通信的设计与实现[J]. 微机发展, 2004, 14(8): 64-66.

(上接第 205 页)

- el - clocked circuits[J]. IEEE Transactions on Computer - Aided Design of Integrated Circuits and Systems, 1999, 18: 1249-1264.
- [9] Shenoy N, Brayton R, Sangiovanni - Vincentelli A. Retiming of circuits with single phase transparent latches, Computer Design: VLSI in Computers and Processors[C]//ICCD'91. Pro-

- ceedings 1991 IEEE International Conference. Cambridge, Massachusetts, USA: IEEE Computer Press, 1991.
- [10] Mullins R, Taylor G, Robinson P, et al. Point to Point GALS Interconnect[C]//ASYNC '02: Proceedings of the 8th International Symposium on Asynchronous Circuits and Systems. Manchester, UK: IEEE Computer Press, 2002.