

基于 Cortex-M3 的汉盲翻译 SoC 设计^①



毛 扬¹, 梁宏博², 邹成洋³, 毛方涛², 吴新丽⁴, 杨文珍²

¹(浙江理工大学 信息科学与工程学院, 杭州 310018)

²(之江实验室 类人感知研究中心, 杭州 311121)

³(浙江理工大学 计算机科学与技术学院, 杭州 310018)

⁴(浙江理工大学 机械工程学院, 杭州 310018)

通信作者: 杨文珍, E-mail: ywz@zhejianglab.edu.cn

摘 要: 汉盲翻译是一种将中文文本自动翻译为对应的盲文数据的过程. 在嵌入式环境下, 汉盲翻译的速度较慢, 难以达到复杂环境下的实时性需求. 为此设计出专用的汉盲翻译 IP 核, 通过实现逆向最大匹配分词算法、汉盲转换, 最终得到准确的盲文数据. 为了验证设计的合理性, 以 Cortex-M3 为微处理器构建 SoC, 搭载串口、LCD 驱动和汉盲翻译 IP 核, 并使用 FPGA 实验平台进行功能验证和性能测试. 测试结果表明, 该 SoC 可准确进行汉盲翻译, 翻译速度达 5 079.37 B/s.

关键词: Cortex-M3; 片上系统; 现场可编程逻辑门阵列; 逆向最大匹配

引用格式: 毛扬, 梁宏博, 邹成洋, 毛方涛, 吴新丽, 杨文珍. 基于 Cortex-M3 的汉盲翻译 SoC 设计. 计算机系统应用, 2023, 32(10): 132-139. <http://www.c-s-a.org.cn/1003-3254/9256.html>

SoC Design of Chinese-braille Translation Based on Cortex-M3

MAO Yang¹, LIANG Hong-Bo², ZOU Cheng-Yang³, MAO Fang-Tao², WU Xin-Li⁴, YANG Wen-Zhen²

¹(School of Information Science and Engineering, Zhejiang Sci-Tech University, Hangzhou 310018, China)

²(Humanoid Perception Research Center, Zhejiang Lab, Hangzhou 311121, China)

³(School of Computer Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310018, China)

⁴(School of Mechanical Engineering, Zhejiang Sci-Tech University, Hangzhou 310018, China)

Abstract: Chinese-braille translation refers to the process of automatically translating a Chinese text into corresponding braille data. In an embedded environment, the speed of Chinese-Braille translation is relatively slow, and it is difficult to meet the real-time requirements in complex environments. Therefore, an IP core for dedicated Chinese-braille translation is designed, which can achieve accurate braille data by implementing a reverse maximum matching segmentation algorithm and Chinese-braille conversion. In order to verify the rationality of the design, the SoC is constructed with Cortex-M3 as the microprocessor, which is equipped with serial ports, LCD drivers, and an IP core for Chinese-braille translation. Functional verification and performance testing are carried out by using FPGA experimental platform. The test results show that the SoC can accurately perform Chinese-braille translation at a speed of 5 079.37 B/s.

Key words: Cortex-M3; system on chip (SoC); field programmable gate array (FPGA); reverse maximum matching

随着人口的不断增长、人口老龄化问题日趋严重, 视障人群的数量也处于不断增长阶段. 据 2019 年统计, 我国中度视力障碍人群有 4 592 万, 重度视力障碍人群

有 467 万, 失明人群有 869 万, 与 1990 年相比增加了 64.35%^[1]. 视障群体缺少重要的视觉感官, 只能通过触觉与听觉来获取信息, 很难像正常人那样获得信息时

① 基金项目: 国家重点研发计划 (2021YFF0600203); 之江实验室科研攻关项目 (K2022PG1BB01, 2022MG0AC04)

毛扬和梁宏博为共同第一作者

收稿时间: 2023-03-08; 修改时间: 2023-04-07, 2023-04-28; 采用时间: 2023-05-06; csa 在线出版时间: 2023-07-21

CNKI 网络首发时间: 2023-07-21

代带来的便利。盲文信息无障碍化建设是为解决视障群体的生活和学习不便问题,使其在任何环境下都可方便平等且无障碍地进行信息获取。在此需求下,出现了以计算机实现盲文数字化的方法,实现文字与盲文之间的翻译作用。现存的汉盲翻译方法多以软件形式实现,在嵌入式环境下,受限于处理器的算力,难以达到实时的汉盲翻译效果。

对于数据密集型计算,常用的加速方法是使用专用的硬件加速器(如 GPU 和 FPGA),通过对数据的并行处理来提高运算的效率。如基于神经网络算法模型设计出的专用 CNN 硬件加速器,因其具有优异的性能而被广泛应用于图像处理、语音识别等应用中^[2],也可用于自然语言处理领域。硬件也可用于实现多模式匹配算法,通过并行流水线结构设计和折叠窗口的 BF 计算来提高算法的执行速度和准确率^[3],可应用于文本搜索、网络安全等领域。在盲文翻译领域,Zhang 等人^[4]提出过一种基于马尔可夫系统的在硬件上实现的盲文翻译器,可实现英文与二级盲文的相互转换;该方法利用 FPGA 的可重构性和并行性,加快了翻译的速度,但该方法只对英文进行翻译,无法翻译其他语言。这项成果为盲人朋友提供了更加便捷的英文盲文互译工具,同时也为硬件实现盲文翻译提供了新的思路和方法。

片上系统(system-on-chip, SoC)是一种将用户所需功能进行集成、在单一芯片上实现复杂功能的特殊集成电路^[5],由可复用的 IP 核(intellectual property core)^[6]组成,具有广泛的应用领域。为促进我国盲文信息无障碍建设,提高嵌入式环境下汉盲翻译的速度,本文在逆向最大匹配分词算法^[7]的基础上设计了一种专用的汉盲翻译 IP 核,并基于开源 Cortex-M3 软核^[8]构建出汉盲翻译 SoC,通过数据的并行处理和减少程序运行时的取指、译码等控制逻辑,提高汉盲翻译算法的执行效率。为测试设计的合理性,本文使用集成开发工具 Vivado 和 FPGA 实验平台构建出完整的汉盲翻译 SoC,并进行了测试实验和性能分析。

1 汉盲翻译的发展情况

由于世界上的盲文体系均是基于布莱叶盲文构建的,盲文在表现形式上具有一致性。盲文以“方”作为最小的表意单位,一方盲文包含 6 个点位,通过调整每个点位的凹凸情况,可形成 2^6 种排列组合,形成基本的盲文显示符号。如图 1 所示为盲文的满方和空方示意图。

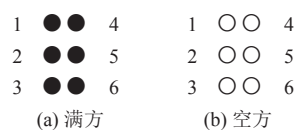


图 1 盲文示例

虽然盲文的表现形式相同,但因为语言的差异,各个国家在文字与盲文之间制定了不同的转换规则。若是字母文字,由于字母与盲文存在着映射关系,盲文转换较为简单。但对一些文字繁多、表意复杂的语言,文字与盲文之间无法建立起唯一映射,盲文转换困难。汉字就属于此类情况,仅常用汉字就有 3 500 个。汉字与盲文之间的转换较为复杂,常使用的是以汉字-拼音-盲文的方案实现盲文翻译^[9]。一般情况下,2-3 方盲文即可表示一个汉字,其中声母、韵母各占一方盲文,部分情况下还需增加一方表示声调。但是,汉字常出现多音、多调、同音不同字的情况,这需要使用算法进行甄别。

汉语没有自然断点,不同的断句常会有不同的语义和读音,因此汉字分词是汉盲转换的重要环节。现有的汉字分词算法按实现原理可分为:基于词典的机械匹配方法^[10]、基于概率统计模型方法和基于机器学习的语义理解方法。Wang 等人^[11,12]提出一种基于盲文语料库的汉盲转换方法,该方法建立起大规模的盲文语料库,以机器学习来训练 LSTM 模型,具备较高的分词准确率。并使用 bi-gram 语言模型对汉字进行标调,建立起汉字标调语料库。该方法无需考虑汉盲转换中的流程和逻辑,转换效果优于采用纯盲文语料库的方法,缺点是标准库搜集困难、校验繁琐。Zhang 等人^[13]自建了通用盲文语料库,结合 HanLp 词典和汉语-盲文词语料库(一种用于生成单元语言模型的盲文分词和连接词典),利用 n-gram 语言模型设计实现了集成汉语、盲文分词和连接规则的汉盲互译系统,汉盲转换准确率达 95.01%。目前,汉盲翻译算法多使用软件实现,以硬件方式实现汉盲转换的方向仍是一片空白。为填补该空白,提高在嵌入式环境下的汉盲翻译速度,本文设计了专用的汉盲翻译 SoC,设计方法详见下文。

2 系统结构设计

2.1 SoC 整体框架

本文设计的 SoC 整体框架如图 2 所示,该设计由

Cortex-M3 软核、串口通讯、汉盲翻译 IP 核、LCD 驱动显示等部分组成. 其中 Cortex-M3 主要负责执行软件部分的功能, 同时负责调用外设运行、与外设进行数据传输、进行初始化配置等工作. 串口通信负责接收上位机传输的文本数据和控制指令, 并将翻译得到的数据传输给上位机或点阵显示模块. 汉盲翻译 IP 负责接收中文的 Unicode 编码、进行中文分词与汉盲转换, 实现汉盲翻译功能. 为方便实验人员观察, 本文通过 LCD 驱动模块将盲文数据在 LCD 屏幕上以点阵的形式显示出来.

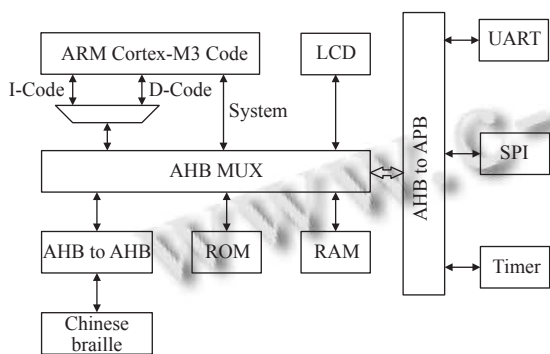


图2 汉盲翻译 SoC 结构框图

Cortex-M3 是 ARM 公司旗下的一款 32 位开源处理器, 该处理器基于哈佛总线架构进行设计, 指令总线 I-Code 和地址总线 D-Code 公用一个地址空间^[14]. 在处理器中已经制定了粗略的存储器映射, 如图 3(b) 所示, 相关的开发过程必须在指定的地址空间范围内进行. 如 Code 区的地址范围为 0x00000000-0x1FFFFFFF, 其中仅使用了 0x00000000-0x0000FFFF 区间作为指令存储器 ROM, 大小为 64 KB. 此外, 本文使用 0x20000000-0x2000FFFF 范围作为数据存储器 RAM. Cortex-M3 预定义 0x40000000-0x5FFFFFFF 为片上外设地址, 将其划分为两个部分, 分别连接 APB 桥接器 (0x40000000-0x4FFFFFFF) 和 AHB 桥接器 (0x50000000-0x5FFFFFFF)^[15]. UART、SPI、Timer 等部分没有太高的带宽需求, 将其挂载在 APB 总线上; 汉盲翻译属于一种数据密集计算方法, 为提高汉盲翻译模块的计算执行效率, 本文将汉盲翻译模块和 LCD 显示模块挂载在 AHB 总线上, 如图 3(a) 所示为外设的地址映射. 为平衡汉盲翻译的效率和 SoC 的功耗, 让不同的模块工作在不同的时钟频率下, 实现跨时钟域传输和运行. 其中, 处理器、ROM、RAM、AHB Mux 和 LCD 模块的

时钟域为 50 MHz, Chinese braille 模块运行在 100 MHz 时钟域下.

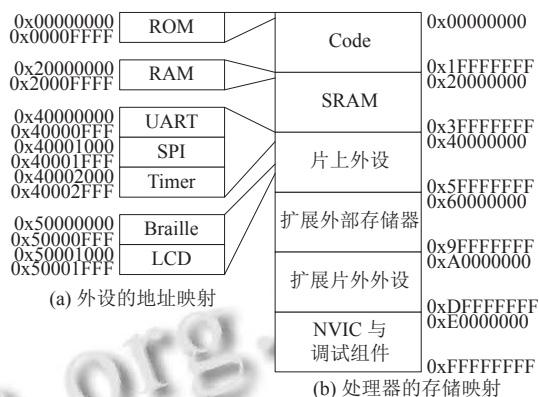


图3 Cortex-M3 地址映射

2.2 汉盲转换 IP 设计

汉盲翻译是一种将汉字转换为 6 点盲文的过程, 是实现盲文无障碍建设的重要环节. 汉盲翻译一般包含中文分词、汉盲转换、符盲转换、点序输出等步骤. 根据汉盲转换的实现步骤, 本文设计出一个汉盲翻译 IP 核, 其结构如图 4 所示. 整个模块由读写缓存、中文分词模块、单字转换模块、多字转换模块、比较模块、Flash 读写模块组成. 当处理器调用该模块时, 经 AHB 总线向写缓存写入文本数据与控制指令; 翻译完成后, 完成信号使能, 处理器由读缓存读取翻译得到的分词结果和盲文点序. 具体的实现原理和步骤见下文.

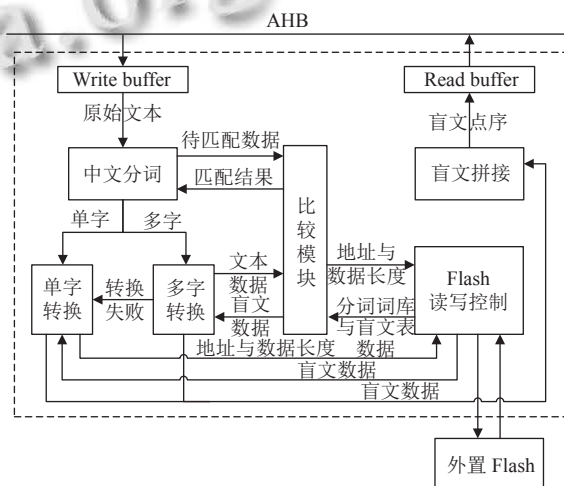


图4 汉盲翻译 IP 结构图

2.2.1 中文分词模块设计

分词是自然语言处理中的重要环节, 直接决定了

汉盲转换的准确性. 考虑到汉语信息重点常置于句末, 且单个词汇的长度决定了该词包含的信息量, 本文选取逆向最大匹配算法实现中文分词, 其实现流程如图5所示.

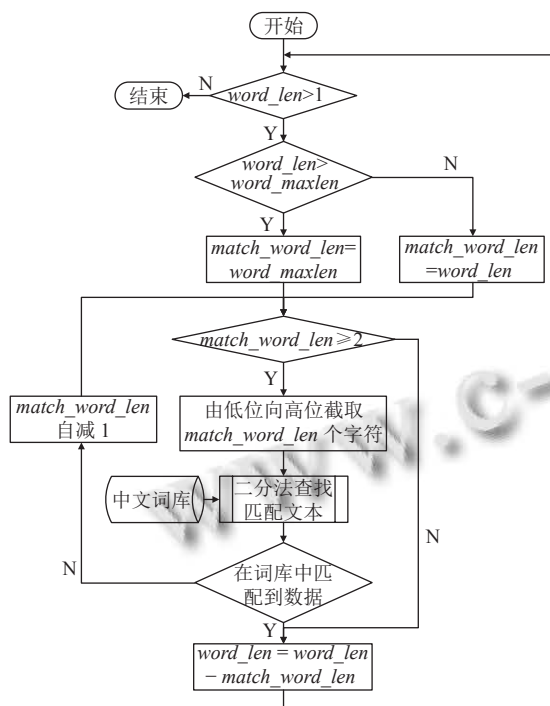


图5 逆向最大匹配分词算法流程

对输入字符串进行初步分割后, 获得不定长的中文与非中文字符串; 其中中文文本需要进行逆向最大匹配分词处理, 记录待分词文本的长度 $word_len$. 设定最大匹配长度 $word_maxlen$, 若 $word_len \geq word_maxlen$, 将文本由低位向高位分割出 $word_maxlen$ 个字符进行匹配; 若 $2 \leq word_len \leq word_maxlen$, 则将当前文本长度作为匹配长度; 若 $word_len = 1$, 无需进行匹配, 直接进行盲文转换. 将分割出的多字字符串与分词表中的字符进行对比, 若匹配成功, 则在该字符串前插入空格, 记录待匹配字符串长度 $match_word_len$, 并将字符串送入多字转换模块; 否则, 将该匹配文本的长度减1, 继续进行匹配, 直到剩下单字符, 此时认为该字符单独成字, 将其送入单字转换模块. 经由上述步骤, 即可实现中文分词功能.

逆向最大匹配分词依赖于分词词库, 本文使用的是一种基于隐马尔可夫模型建立的分词词库, 为提高算法执行的效率, 本文将词库数据以 Unicode 格式进行存储, 根据词条的长度和编码顺序进行分类和排列.

表1所示为分词词库的词条统计, 考虑到盲人不方便摸读过长的词汇, 本文限定词库的最大词长为16 B. 由于分词词库中的词条数目庞大, 若使用顺序匹配的方法, 执行效率极低, 分词的实时性也无法满足需求. 由于词条是按汉字数量和编码顺序进行排列存储, 可以使用二分法进行查找.

表1 分词词库词条统计

词长 (B)	词条数量	词长 (B)	词条数量
4	114 128	12	1 641
6	131 337	14	918
8	84 894	16	385
10	3 563		

2.2.2 汉盲转换模块

经分词处理后, 即可得到不定长度的中文词组和其他符号, 继而进行盲文转换过程. 对于英文字母、数字、标点符号等字符, 由于对应的盲文转换规则较少, 可在转换系统中建立与之映射的盲文编码, 直接使用选择器进行盲文转换. 汉字在转换时需进行汉拼转换, 得到正确的拼音. 但因汉字数量庞大, 且存在一字多音、多调和多音字的情况, 无法建立起与拼音的唯一映射关系; 同时, 拼音转换盲文需要遵循标调规则、声调省写规则、汉字简写规则等规则, 在硬件上实现比较困难. 因此, 本文建立起一个汉字-盲文转换词库, 硬件上以查表法读取词条内容和输出盲文数据.

汉字有单音节和多音节之分, 导致汉字的编码和盲文长度没有直接关联. 为防止系统读取盲文数据时出现冗余或缺失, 在汉字 Unicode 码与盲文数据之间添加 1 B 空间, 写入盲文长度. 如图6所示是部分盲文对照库的数据结构, 虚线框中的数据是 6 点盲文的数量, 虚线框左侧是 3 字词汇的 Unicode 编码, 虚线框右侧是 HEX 格式的盲文点序. 在 Unicode 字符集中, $\backslash x4E00-\backslash x9FA5$ 范围内的编码属于中日韩统一表意文字, 共计 20 902 个词条, 在本文作为盲文转换的单字使用. 异常发音词汇的盲文数据源于开源盲文翻译软件 Liblouis 中的现行盲文方案中的异常发音汉语词汇, 共计 114 099 个词条.

汉盲转换模块分为单字转换和多字转换两个部分. 当进行单字转换时, 根据单字盲文表的初始地址和 Unicode 码数据计算出目标地址, 即可得到单字盲文. 当分词结果为多字时, 需根据分词长度确定目标盲文的地址范围, 通过与盲文表中的中文 Unicode 部分进

行匹配对比,若匹配成功,输出盲文数据和盲文方数;否则,认定该分词结果没有特殊的拼音和音调,将其以单字转换的形式,逐字进行盲文转换。

中文	Unicode 编码	盲文方数															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0026f340	62 53 4e 0b 62 4b	08	98	a2	32	d6	3e	c8	00	00	00	00	00	00	00	00	
0026f350	62 53 4e 0d 4f 4f	07	98	a2	30	a4	ca	40	00	00	00	00	00	00	00	00	
0026f360	62 53 4e 3b 61 0f	08	98	a2	0c	a4	85	18	00	00	00	00	00	00	00	00	
0026f370	62 53 4e a4 90 53	08	98	a2	36	3a	09	9a	00	00	00	00	00	00	00	00	
0026f380	62 53 51 48 95 0b	08	98	a2	32	96	0d	0f	00	00	00	00	00	00	00	00	
0026f390	62 53 51 49 68 cd	08	98	a2	36	6e	0d	92	00	00	00	00	00	00	00	00	
0026f3a0	62 53 51 b2 95 0b	08	98	a2	3e	4e	0d	0f	00	00	00	00	00	00	00	00	
0026f3b0	62 53 51 b7 62 18	08	98	a2	38	3c	83	39	00	00	00	00	00	00	00	00	
0026f3c0	62 53 51 b7 67 aa	09	98	a2	38	3c	8a	2d	80	00	00	00	00	00	00	00	
0026f3d0	62 53 52 4d 7a d9	07	98	a2	28	94	ce	40	00	00	00	00	00	00	00	00	

图6 盲文对照库的数据结构

2.2.3 Flash 读写模块

在执行分词和汉盲转换时,需要使用较大的分词词库和盲文查找表,共计需要占据 3.65 MB 存储空间。本文使用外置的 Flash 作为存储器,型号为 W25Q128,存储大小为 16 MB,对外接口为 SPI 接口。当比较模块接收到分词和多字盲文转换的文本数据时,根据文本的长度确定地址范围,以二分法确定中间地址,并驱动 Flash 读写模块读取数据进行匹配。若匹配成功,输出成功信号和目标地址;否则进行地址转换,直到超过地址范围,输出匹配失败信号。图 7 所示为 Flash 读写模块的结构框图,其中 *SPI_Ctrl* 负责接收控制指令、传输地址和数据,通过控制指令决定对 Flash 进行数据读写或擦除功能。由于读写 Flash 时,只需传输首地址,然后持续拉低 CS 信号即可实现数据的连续读写,一次最多可对一页 (256 字节) 进行操作。为保证数据的完整性,设置了 *Tx_fifo* 和 *Rx_fifo*, 分别缓存发送的数据和接收的数据,两个 fifo 位宽均为 8 bit,深度为 256 个。*SPI_Shift* 接收到 *SPI_Ctrl* 传递的 Flash 控制指令和地址数据,并将这些数据移位传输至 Flash: 若为写操作,需要从 *Tx_fifo* 中获取数据进行移位传输;若为读操作,从 MISO 数据线获得串行数据,进行串并转换后写入 *Rx_fifo*, 并等待外部读取。同时, *SPI_Ctrl* 也负责接收 *Tx_fifo* 和 *Rx_fifo* 的当前状态,包括空、满信号、写入的数据量等。

2.2.4 盲文拼接模块

经盲文转换后的盲文数据是 16 进制的无符号数,不具备可读性。为此,本文设计单独的盲文拼接模块对盲文数据进行处理。该模块有两个功能,一是在分词结果对应的盲文之间添加一个空方,表示在盲文信息中

的断句,确保盲文数据能显示准确的含义;二是按顺序拼接由盲文转换模块输出的盲文数据,最终得到正确的汉盲翻译结果。

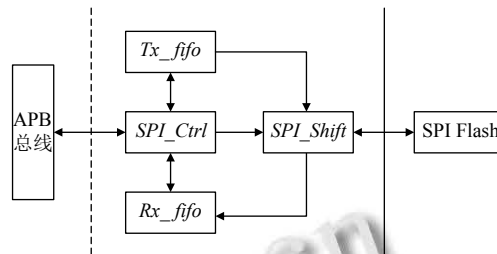


图7 Flash 读写模块框图

3 仿真与实验结果

3.1 汉盲转换 IP 的仿真结果

前文详细介绍了汉盲翻译 SoC 的设计方案,根据该方案,本文在 Xilinx 公司提供的 Vivado 集成开发工具上搭建出 SoC,与 Modelsim 仿真工具配合进行综合仿真和时序验证。仿真结果如下。

当微处理器调用汉盲翻译模块时,通过 AHB 总线向翻译模块的寄存器中写入数据。本文设置了 3 组寄存器,分别为控制寄存器、写数据寄存器和读盲文寄存器。其中控制寄存器除输入控制指令与中文长度外,还负责输出翻译得到的盲文方数。如图 8 所示是 AHB 总线工作的时序图。当 HSEL 使能时,选中汉盲翻译从机;此时 HREADY 和 HWRITE 同时使能,向 HADDR 指向的寄存器写入中文数据,写入完成后向控制寄存器写入控制指令。在此次仿真中,共向写数据寄存器中写入了 4 个 32 位数据 (即 8 个 Unicode 汉字编码),写入的控制指令为 0x00010008, 0-7 位为中文字长,第 16 位为汉盲翻译使能位。翻译完成后,当 HWRITE 为低电平时读出控制指令,此时控制寄存器中的翻译使能位失能,同时向 8-15 bit 写入翻译得到的盲文方数,最终得到了 15 方盲文。若需要获取盲文数据, HADDR 指向读盲文寄存器, HWRITE 为低电平时,从该寄存器中获取盲文数据。

汉盲翻译从机的仿真效果如图 9,由上到下分别为分词模块、多字转换模块、单字转换模块和盲文拼接模块的仿真时序。当 *match_en* 使能且有 Unicode 数据输入时,开始执行分词匹配。若匹配失败, *match_error* 使能一个时钟周期,同时将 *match_word_in* 左移 16 位 (即去除一个左侧汉字的 Unicode 码),继续进行匹配。当匹

配成功时, *match_result*使能一个周期, 同时进行盲文转换. 当分词结果有多个汉字时, 先进行多字转换, 将分词结果与盲文对照表的数据进行比较; 若未找到对应的盲文词条, *braille_error_n*使能一个周期, 进行单字转换. 此时, 将分词结果每 16 bit 进行一次汉盲转换,

最终得到盲文数据, 图 9 中得到的是 6 方盲文数据 0xe2a436510, *braille_result_1*使能一个周期, 并继续执行盲文拼接操作. 将转换得到的盲文与先前获取的数据进行拼接, 并在两组盲文之间添加空方, 得到带分词效果的盲文数据.



图 8 AHB 总线数据读写

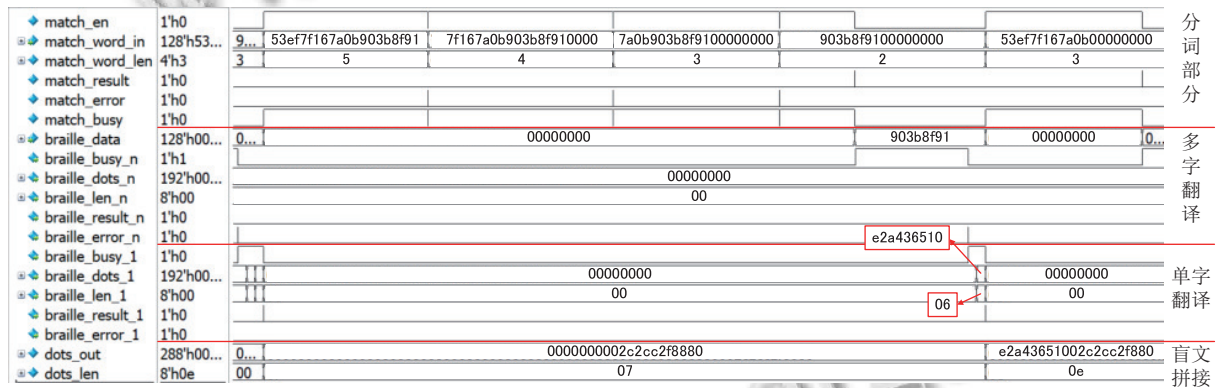


图 9 中文分词及汉盲翻译仿真

为进行 Flash 仿真, 本文使用了 W25Q128 的仿真模型作为 Flash 读写模块的从机. 由于在汉盲翻译过程中, 对 Flash 主要执行的是读操作, 故本文仅对读逻辑进行时序仿真, 如图 10 所示. 当启用 Flash 读写模块读功能时, 由外部输入地址 *rd_addr* 和起始信号 *rd_flash_en*, 随后拉低 *spi_cs*, 开始进行 SPI 传输: 首先发送 8 位的读指令 03h 和 24 位地址 1e70d0h, 每发送 8 bit *tx_byte_done* 使能一个周期; 发送完成后 Flash 经 *spi_miso* 传回数据, 并由移位寄存器 *wr_data* 进行移位存储, 每接收 8 bit 时 *rx_byte_done* 使能一个周期, 同时将数据写入 *Rx_fifo*. 当读取完数据后, 拉高 *spi_cs* 停止 SPI 数据传输, 同时读取 *Rx_fifo* 缓存的数据. 经对比分析, 图 10 中显示读出的 16 字节数据与仿真模型中的数据一致.

3.2 实验测试结果

为验证本文设计的汉盲翻译 SoC 的合理性和可行性, 本文使用 Xilinx 公司型号为 xc7a200tfg484 的 FPGA, 在其中构建和运行 SoC. 为获取汉盲翻译的结果, 除构建汉盲翻译 IP 外, 在其中搭载了串口和 LCD 驱动显示接口, 并进行了多次实验. 如图 11 所示是以汉字“一种半定制的现场可编程逻辑门电路”的 Unicode 编码作为输入的翻译结果, 用 LCD 屏幕进行显示, 空方表示盲文分词的位置. 结果表明, 汉盲翻译结果无误.

为确定汉盲翻译的速度, 本文进行了相关测试. 当前主流的盲文显示装置一次可显示 40 或 80 方盲文^[16], 以 80 方为例可显示 50–100 B 左右的汉字信息. 因此,

本文在 50–100 B 的长度中选取了 5 组不同规模的中文文本作为样本输入, 并进行多次测试, 得到平均翻译时间和翻译速度, 详见表 2。表中, 样本 1–样本 4 是从小学三年级语文课文中随机摘取的文本数据, 分为两个数量规模进行测试。根据表中数据, 发现不同样本之间的翻译速度略有差异。经过分析, 这是由于每次翻译过程中根据标点符号截取的中文字符串长度不同导致的, 理论上连续的中文字符串越长、单次翻译所需时间也越长。如对分段的 8+8 字文本和连续的 16 字文本,

前者的翻译速度一般高于后者。本文设计的汉盲翻译 IP 核一次最多可翻译 16 字的文本数据, 为验证文本为 16 字时的汉盲翻译速度, 选取了包含 2 条 16 字的文本数据作为样本 5 进行测试, 翻译速度为 5 079.37 B/s, 远低于普通样本的转换速率。文献 [17] 中, 在时钟频率为 168 MHz 的 STM32 平台下运行了汉盲翻译软件程序, 并使用 50 字节的文本数据进行测试, 得到的翻译速度为 217.39 B/s。与其相比, 本文构建的汉盲翻译 SoC 的翻译速率仍有 20 倍以上的提升。

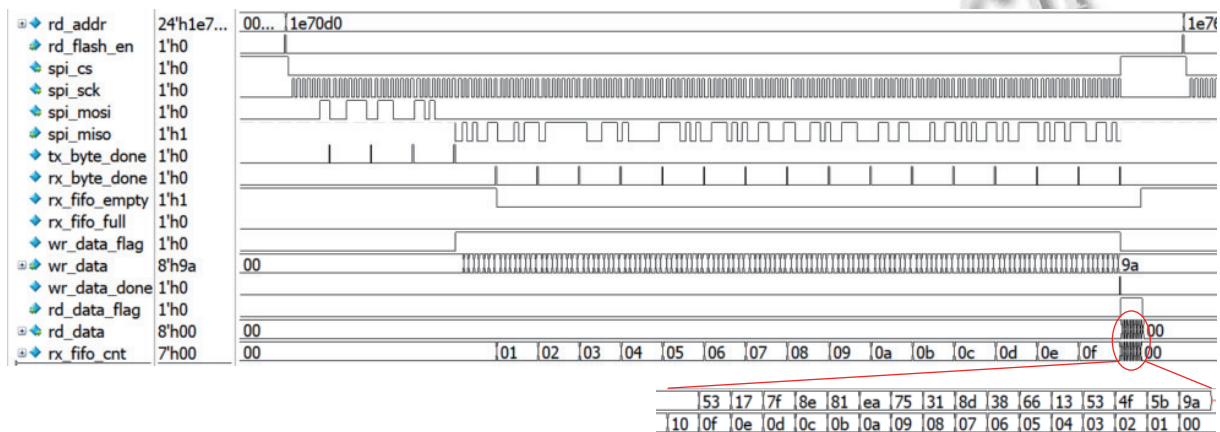


图 10 Flash 读写仿真

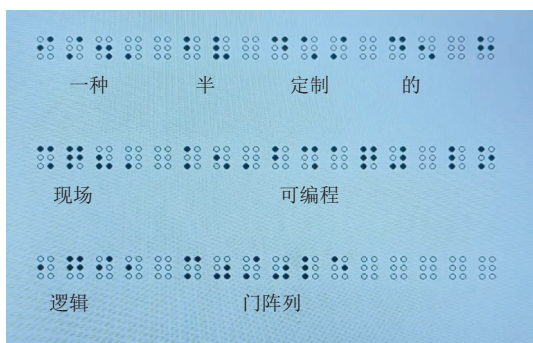


图 11 汉盲翻译效果图

表 2 汉盲翻译速度统计

样本	字符数 (B)	平均翻译时间 (s)	翻译速率 (B/s)
样本1	54	0.0053	10188.68
样本2	58	0.0061	9508.20
样本3	100	0.0102	9803.92
样本4	102	0.0103	9902.91
样本5	64	0.0126	5079.37
文献[16]	50	0.23	217.39

习方面越来越受到重视。盲文数字化的提出和推进, 为信息时代下视障群体的无障碍信息获取提供了有利条件。为推进盲文数字化发展, 扩大盲人群体获取盲文信息的场景, 本文设计出专用于汉盲翻译的 SoC 系统, 实现对中文 Unicode 数据的获取, 以逆向最大匹配分词进行中文分词, 依靠建立的盲文对照表进行汉盲转换, 最终快速且准确地得到盲文数据。本文使用 Xilinx 公司提供的 XC7A200T FPGA 搭建 SoC 并成功运行, 经测试, 汉盲翻译的速度可达 5 079.37 B/s, 可满足嵌入式环境下的需求。该 SoC 使用 Cortex-M3 作为处理器, 具有较强的扩展性, 方便进行后续功能的添加。

致谢

感谢胡庆玲、刘畅、陶文韬、阮刘霞等人在此研究工作中做出的贡献。

参考文献

1 Xu TL, Wang BS, Liu H, et al. Prevalence and causes of

4 结束语

随着社会的不断进步, 视障群体的日常生活和学

- vision loss in China from 1990 to 2019: Findings from the global burden of disease study 2019. *The Lancet Public Health*, 2020, 5(12): e682–e691. [doi: [10.1016/S2468-2667\(20\)30254-1](https://doi.org/10.1016/S2468-2667(20)30254-1)]
- 2 Bouaafia S, Khemiri R, Messaoud S, *et al.* Deep CNN co-design for HEVC CU partition prediction on FPGA-SoC. *Neural Processing Letters*, 2022, 54(4): 3283–3301. [doi: [10.1007/s11063-022-10765-1](https://doi.org/10.1007/s11063-022-10765-1)]
- 3 骆潇, 郭健, 邓敏, 等. 基于FPGA的多模式匹配算法研究与实现. *计算机工程*, 2012, 38(18): 231–233, 237. [doi: [10.3969/j.issn.1000-3428.2012.18.062](https://doi.org/10.3969/j.issn.1000-3428.2012.18.062)]
- 4 Zhang X, Ortega-Sanchez C, Murray I. Text-to-braille translator in a chip. *Proceedings of the 2006 International Conference on Electrical and Computer Engineering*. Dhaka: IEEE, 2006. 530–533.
- 5 Chakravarthi VS. A practical approach to VLSI system on chip (SoC) design: A comprehensive guide. Cham: Springer, 2020.
- 6 Nath APD, Boddupalli S, Bhunia S, *et al.* Resilient system-on-chip designs with NoC fabrics. *IEEE Transactions on Information Forensics and Security*, 2020, 15: 2808–2823. [doi: [10.1109/TIFS.2020.2977534](https://doi.org/10.1109/TIFS.2020.2977534)]
- 7 Zhang LY, Li YZ, Meng J. Design of Chinese word segmentation system based on improved Chinese converse dictionary and reverse maximum matching algorithm. *Proceedings of the 2006 International Workshops on Web Information Systems*. Wuhan: Springer, 2006. 171–181.
- 8 Ge F, Wu N, Xiao H, *et al.* Compact convolutional neural network accelerator for IoT endpoint SoC. *Electronics*, 2019, 8(5): 497. [doi: [10.3390/electronics8050497](https://doi.org/10.3390/electronics8050497)]
- 9 Zhu XY, Bao T. EasyBraille: A translation system for mandarin and braille natural language understanding and machine translation. *Proceedings of the 6th Joint Symposium on Computational Linguistics in China*. Beijing, 2001. 326–331.
- 10 Li ZP, Wang R, Zhang TC, *et al.* Intelligent braille conversion system of Chinese characters based on Markov model. *Proceedings of the 3rd IEEE Information Technology, Networking, Electronic and Automation Control Conference*. Chengdu: IEEE, 2019. 1283–1287.
- 11 Wang XD, Yang Y, Zhang JC, *et al.* Chinese to braille translation based on braille word segmentation using statistical model. *Journal of Shanghai Jiao Tong University (Science)*, 2017, 22(1): 82–86. [doi: [10.1007/s12204-017-1804-x](https://doi.org/10.1007/s12204-017-1804-x)]
- 12 Wang XD, Zhong JH, Cai J, *et al.* CBConv: Service for automatic conversion of Chinese characters into braille with high accuracy. *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility*. Pittsburgh: ACM, 2019. 566–568.
- 13 Zhang JX, Chen, HF, Chen B, *et al.* Design and implementation of Chinese common braille translation system integrating braille word segmentation and concatenation rules. *Computational Intelligence and Neuroscience*, 2022, 2022: 8934241.
- 14 Yiu J. *The Definitive Guide to ARM® Cortex®-M3 and Cortex®-M4 processors*. Amsterdam: Elsevier, 2014.
- 15 ARM Ltd. AMBA specification rev2.0. <http://www.arm.com>. (2022-01-20).
- 16 陈楷闻. 嵌入式盲文数字化系统研究 [硕士学位论文]. 杭州: 浙江理工大学, 2022. [doi: [10.27786/d.cnki.gzjlg.2022.001132](https://doi.org/10.27786/d.cnki.gzjlg.2022.001132)]
- 17 林栋. 面向国家通用盲文方案的盲文转换方法研究 [硕士学位论文]. 杭州: 浙江理工大学, 2022. [doi: [10.27786/d.cnki.gzjlg.2022.000779](https://doi.org/10.27786/d.cnki.gzjlg.2022.000779)]

(校对责编: 孙君艳)