doi: 10.7690/bgzdh.2024.09.014

基于 FPGA 的异步 FIFO 缓存数据溢出控制系统

张　伟
（西安航空职业技术学院人工智能学院，西安 710089）

摘要：为获取更高效、稳定的缓存数据控制方法，设计基于现场可编程门阵列（field programmable gate array，FPGA）的异步 FIFO 缓存数据溢出控制系统。设计存储控制方案，得到基于 FPGA 的系统硬件；建立缓存数据存储溢出模型，得到数据节点剩余能量的最小值最大化求解，在函数模型下判断存储数据是否溢出；设计数据溢出控制算法，得到缓存数据溢出控制系统的软件。分别对数据溢出的监测性能与控制性能进行测试。实验结果表明：使用该方法剩余的能耗较高，可见该方法对于缓存数据的监测与控制性能均较好。

关键词：FPGA；异步 FIFO 存储器；缓存数据；数据溢出控制；存储控制；数据节点
中图分类号：TN06　　**文献标志码：**A

Data Overflow Control System of Asynchronous FIFO Buffer Based on FPGA

Zhang Wei
(*College of Intelligence*, *Xi'an Aeronautical Polytechnic Institute*, *Xi'an* 710089, *China*)

Abstract: In order to obtain a more efficient and stable cache data control method, an asynchronous FIFO cache data overflow control system based on field programmable gate array (FPGA) is designed. A storage control scheme is designed to obtain the system hardware based on FPGA; a cache data storage overflow model is established to obtain the maximum solution of the minimum value of the residual energy of the data node, and the storage data is judged whether to overflow under the function model; a data overflow control algorithm is designed to obtain the software of the cache data overflow control system. The monitoring performance and control performance of data overflow are tested respectively, and the experimental results show that the remaining energy consumption of this method is higher, which shows that this method has good monitoring and control performance for cache data.

Keywords: FPGA; asynchronous FIFO memory; cache data; data overflow control; storage control; data node

0　引言

存储器内通常携带着大量的数据与程序，基于这种记忆功能，存储器设备成为了现代计算机最重要的组成部分，在数据采集、存储以及处理等方面均发挥着重要作用，甚至可以直接影响计算机性能。在这个过程中，不同的时钟系统之间需要更稳定快捷的数据传输方式，集成电路技术也需要更低功耗；此时，异步 FIFO 存储器就成为了这些存储器中最为简便、快捷的一种。它可以解决时钟偏差和时钟延迟的问题，并广泛应用于混杂模块之间的通信以及多频率的芯片设计，具备低功耗、模块化、高效快捷的优点；因此，异步 FIFO 是较为常见且应用价值较大的一种存储器。然而，任何一种存储器的存储空间都是有限的，一旦数据收集的频率过快，就会出现数据存储空间占满的问题，造成数据存储溢出[1]。一旦数据存储溢出，该存储器就无法继续保存数据，甚至导致该计算机无法正常工作；因此，需要对这类现象进行控制。现场可编程门阵列（FPGA），可以被当作集成电路中的一种半定制电路。这类电路可编辑性能较差且运行速度较慢，但是造价便宜，且功能更加丰富，在小批量的系统中具备较高的系统集成度，且可靠性极高。笔者设计一种基于 FPGA 的异步 FIFO 缓存数据溢出控制系统，在建立系统框架之后，详细设计了该系统的硬件与软件部分，并对该系统进行了测试。

1　建立系统框架

为保证笔者设计的异步 FIFO 数据存储器可以在数据溢出时及时控制，需在结合 FPGA，在原有的异步 FIFO 数据存储器内设置一个状态判断逻辑窗口，如图 1 所示。

在笔者的系统框架中，使用编码器与译码器作为决定读写地址的工具，数据经过输入窗口后，进入 RAM，与读写时钟汇合，在阈值寄存器的辅助下，在输出窗口输出数据[2-3]。时钟读写窗口会通过计数

收稿日期：2024-05-22；**修回日期：**2024-06-22
第一作者：张　伟（1990—），男，陕西人，硕士。

器连接状态判断窗口，该窗口可以判定一个数据存储器的状态，包括“内存已满”“内存将满”“内存未满”。

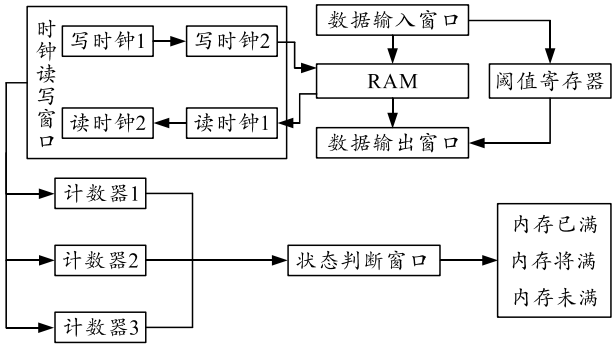

图 1 系统框架

2 基于 FPGA 的系统硬件设计

2.1 存储控制方案设计

高速传输的数据通常由源数据产生，在经过数据输入接口以及数据输出接口后，到达目标设备。在这个过程中，想要保证数据的高效稳定传输，存储系统的结构是重中之重[4-5]。其中，源设备与目标设备之间的数据传输速率是至关重要的一个环节。笔者选择 FPGA+PC 作为服务器或者计算机的存储控制方案。其基本框架如图 2 所示。

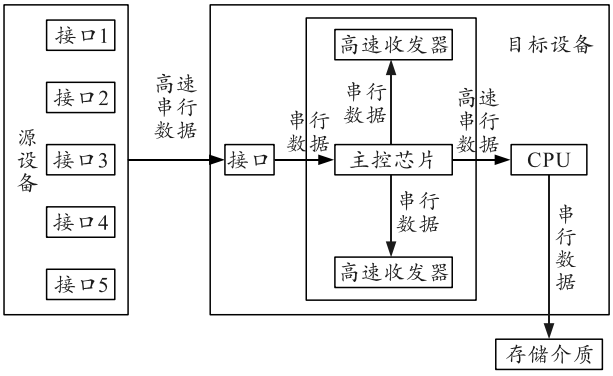

图 2 存储控制方案

上图中，存在源设备、目标设备和存储介质 3 个实体。在基于 FPGA 的存储控制方案中，设备和目标设备是核心架构。将高速串行数据以及串行数据通过上图所示的设备连接，并在硬件上传输[6-7]。源设备中连接了多个接口，首先将其通过高速串行数据传输到目标设备的接口部位，然后经过串行数据传输到主控芯片，主控芯片会与 2 个高速收发器相连，同时还会经过高速串行数据传输到目标设备的 CPU 中，最后通过串行数据传输到存储介质中。

2.2 主控芯片

当前的异步 FIFO 存储器使用 FPGA 作为高速数据收发器件；因此，该硬件系统内的存储器需要通过定制的电路作为特定装备，且完成设备的改造后，不能随便更改，否则会导致硬件损坏。通过该存储控制系统的核心器件，笔者设计的系统能够具备较为强大的原型验证功能、数字信号处理功能、嵌入式功能、物理层通信功能、可重构计算功能、协处理功能等[8-9]。其可以使用更多的逻辑资源以及高速互联功能进行原型验证；同时，还可利用大量的存储资源进行并行计算，在内置的硬核处理器下，完成串行总线协议的实时处理，并基于 FPGA 的现场编程功能，随时切换自身的计算模型，使主控芯片更具适应性。在并行多核计算功能下，完成对于数据的高效传输。

3 缓存数据溢出控制系统软件设计

3.1 建立缓存数据存储溢出模型

想要对异步 FIFO 缓存数据的溢出进行控制，就要建立一个能够判定缓存数据存储溢出的模型。设定在某数据库中，数据节点的数量为：

$$N_s = 1/(|V_m| - p_k)。 \tag{1}$$

式中：N_s 为数据节点的整体数量；V_m 为数据节点集合内的个体数量；p_k 为网络节点[10-11]。

将这些数据节点进行预处理，同时获得部分数据存储溢出的判别式：

$$\sum_{i=1}^{n} f_i \leqslant \sum_{j=1}^{|V_m|-p_k} h_p \Big/ \sum_{i=1}^{|V_m|-p_k} h_k + 1。 \tag{2}$$

式中：f_i 为存储空间内溢出的数据量；h_p 为第 p 个存储节点；h_k 为单位存储节点内溢出的数据量。对数据进行重分布处理，计算数据节点上剩余的能量最小值。设置数据节点 B，令 $B=\{B_1, B_2, \cdots, B_n\}$，其中每个子集都是数据节点内的单位溢出数据[12]。此时属于数据节点的第 i 个溢出数据 B_i 被分配到了存储节点 C_j 上，同时其他溢出数据也都寻找到了相应的存储节点，则可在此时建立一个存储节点 C 的集合 $C=\{C_1, C_2, \cdots, C_n\}$。

在计算重分布路径的能耗时，得到数据节点剩余能量的最小值最大化：

$$\max_{L_i} \min_{1 \leqslant i \leqslant n} P_{N_i}。 \tag{3}$$

式中：P_{N_i} 为数据节点 N_i 的剩余能量；L_i 为节点重分布路径集的子集，它来源于 $L=\{L_1, L_2, \cdots, L_n\}$。

对于数据节点存储路径与能量的转化，可以建立一个函数模型：

$$h\left(\alpha_p,\beta_p\right)=\begin{cases}2\alpha_p,\alpha_p<\beta_p/2\\ \alpha_p+\beta_p/2,\alpha_p\geqslant\beta_p/2\end{cases}。\tag{4}$$

式中 $h(\alpha_p,\ \beta_p)$为针对数据节点存储路径以及能量转化的函数，其中，α_p 为存储节点容量，β_p 为能量转化容量[13-14]。通过上述转化函数，可对数据存储的溢出进行判定，当函数值增大到一定程度时，表示该缓存数据达到存储器的极限。通过该函数就可得到缓存数据存储溢出模型。

3.2　设计数据溢出控制算法

在本文中使用数据融合以及数据转存的方式，对缓存数据的溢出进行控制，结合数据节点与存储节点，可以设定其融合过程如图 3 所示。

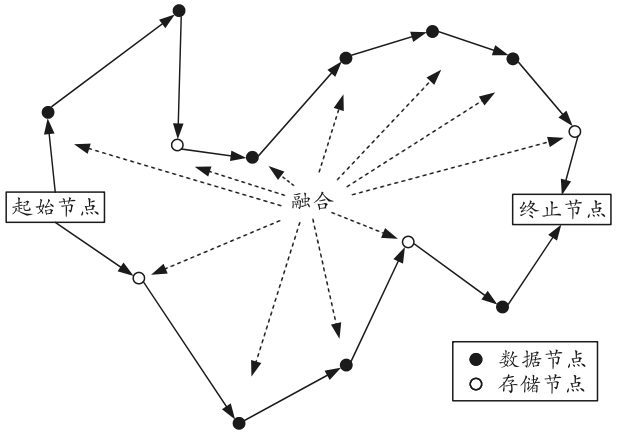

图 3　数据融合过程

结合上图所示的数据融合，通过相关程度获取数据节点的位置。在拥有足够多的数据融合节点时，可以得到融合节点的数量为：

$$N_m=\sum_{i=1}^{n}P_n\times R_n\Big/\sum_{i=1}^{n}\left(P_n-R_n\right)。\tag{5}$$

式中：N_m 为在融合过程的起始节点到终止节点内部，所有待融合节点的数量；P_n 为数据节点的数量；R_n 为存储节点的数量[15]。

据此可以得到融合和转存后剩余的能量值：

$$E_k=E_i-\sum_{j=1}^{n}k_j。\tag{6}$$

式中：E_k 为融合和转存后剩余的能量；E_i 为初始能量；k_j 为第 j 个节点在该过程中的能耗。结合上述公式可对超出存储器极限的缓存数据溢出过程进行控制。

4　实验研究

4.1　实验准备

为测试基于 FPGA 的异步缓存数据溢出控制系统的性能，设计如下实验：1) 通过数据重分布算法的性能对比，测试该系统对于数据溢出监测的能耗；2) 通过数据转存算法的性能对比，测试系统对于数据溢出的控制效果。该实验装置只需使用个人 PC 机，内存为 64 G，硬盘空间为 300 T，整体的实验时间为 36 h。整体的硬件平台搭建好以后，需对以上 2 个功能模块分别进行测试，此时的集成开发环境为 Vivado，将已经生成的文件加载进 FPGA 模块，并对该模块进行初始化。经过初始化处理后的缓存数据处于高电平周期，生成的数据流均为 16 Gbps。在测试系统性能时，将笔者设计的 FPGA 方法与几种现有的方法进行对比，如基于虚拟存储层的数据控制方法、基于流量延时调度机制的数据控制方法、基于模糊控制的数据控制方法等，以测试系统在数据溢出前的检测能力以及对数据溢出的控制能力为目标。

4.2　数据溢出监测性能测试

在实验平台上，设置多个数据节点，每个数据节点均由算法自动生成，且所有数据都可以互通。为保证实验时不会破坏仪器，本次试验所使用的数据量需要在超过异步 FIFO 存储器本身的容量的基础上，小于剩余存储节点的总体数据量。以溢出数据量为变量，分别测试其余数据节点的剩余能量，得到的数据结果如图 4 所示。

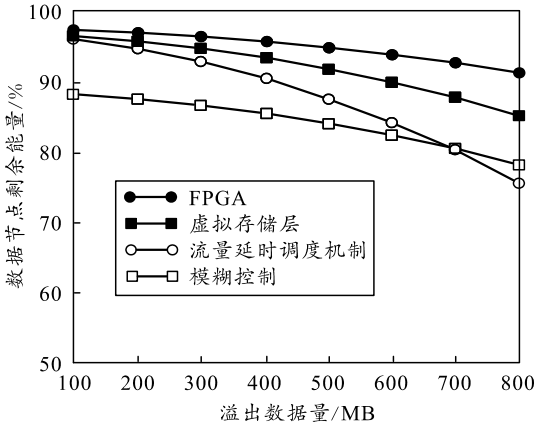

图 4　不同数据溢出量下系统监测能耗

图中所有节点剩余能量均为其单位溢出数据量下的最小值。FPGA 方法下，当溢出数据量为 100 MB 时，数据节点的剩余能量约为 97.1%，虚拟存储层、流量延时调度机制以及模糊控制 3 种方法在该溢出数据量下的数据节点剩余能量分别为 96.9%、96.4%、89.1%。随着溢出数据量的增加，4 种方法的数据节点剩余能量百分比也在不断下降，当溢出数据量提升至 800 MB 时，4 种方法的数据

节点剩余能量百分比分别为 90.8%、85.2%、76.7%、78.8%。在以上 4 种方法中，随着数据溢出量的增加，系统剩余能量下降幅度最小的是笔者设计的 FPGA 方法，下降幅度最大的则是流量延时调度机制方法。

在试验中，设置 r 表示系统数据节点的溢出量，设置 m 表示系统所剩余的存储空间，使用 r/m 作为单位存储空间内分配溢出数据的性能参数，分别测试不同 r/m 下系统剩余能量的百分比，如图 5 所示。

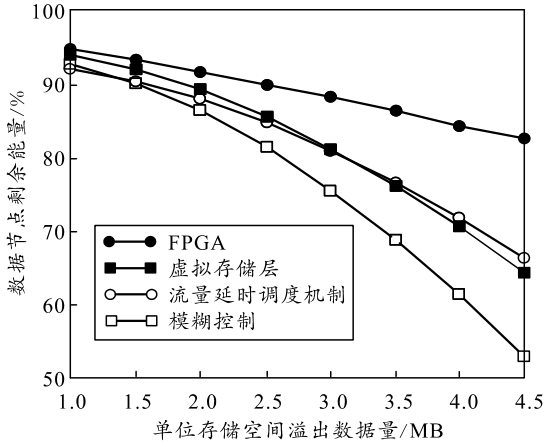

图 5 不同 r/m 下系统监测能耗

如上图所示，在单位存储空间所分配的数据溢出量性能测试中，随着 r/m 逐渐增加，数据节点剩余能量的百分比正在不断下降。在 FPGA 方法下，r/m 为 1 时的数据节点剩余能量约为 95.0%，但是当 r/m 为 4.5 时，其剩余能量只有 83.1%。在剩余的 3 种对比方法中，r/m 为 1 时的剩余能量分别为 94.3%、92.8%、92.9%，当 r/m 为 4.5 时，其数据节点剩余能量则分别为 64.9%、66.7%、52.6%。

综合上述 2 个不同自变量下数据溢出监测能耗的测试可知，随着数据溢出量的增加，无论是整体的系统能耗，还是单位节点下的系统能耗，均在不断增加，剩余的能量百分比在减少。而文中方法与 3 种传统方法的对比中可知，本文中方法数据节点剩余能量在任何条件下均为 4 种方法下的最大值，可见该方法优于其他 3 种方法。

4.3 数据溢出控制性能测试

在测试文中系统针对数据溢出控制的能力时，需将数据的转存能力作为衡量标准。在该实验中，设置数据节点的数量为 50，且每个数据节点溢出的数据量基本相同，每个存储节点剩余的存储空间也相等。计算溢出数据相关系数：

$$\rho=1-M_R/r \text{。} \tag{7}$$

式中：ρ 为溢出数据相关系数；M_R 为数据节点完成数据融合后溢出的数据量；r 为系统数据节点的溢出量，此时 r/m=1。分别取溢出数据相关系数为 0.5 和 0.7 时的值，并获取不同溢出数据下系统的转存性能。

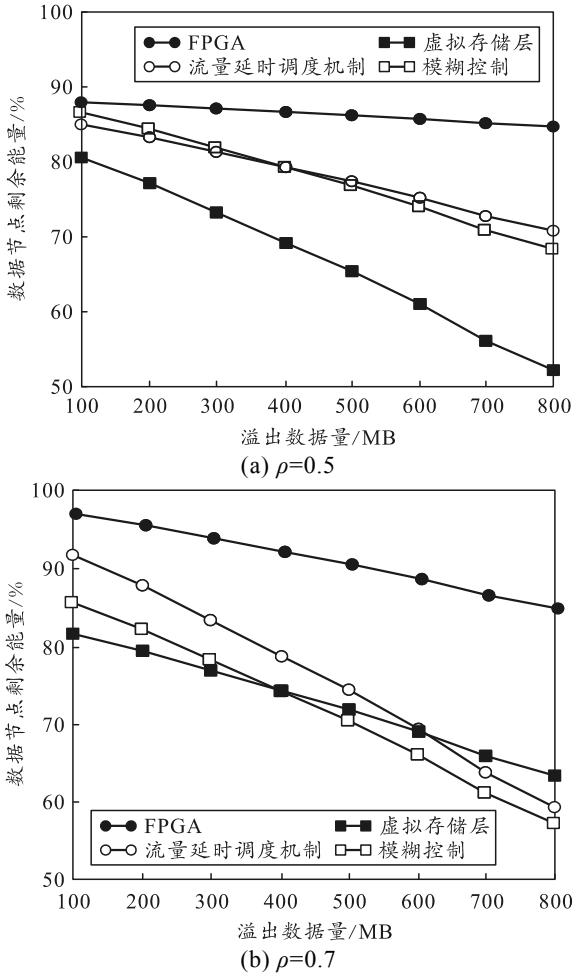

图 6 不同数据溢出量下系统转存能耗

当溢出数据相关系数为 0.5 时，随着溢出数据量的增加，数据节点剩余能量也随之减小。在 FPGA 方法中，数据节点剩余能量的下降幅度极小，仅仅由 88.7%下降至 84.7%。虚拟存储层的下降幅度最大，由 80.9%下降至 52.3%。流量延时调度机制与模糊控制 2 种方法的数据节点剩余能量下降幅度相似，均小于虚拟存储层方法，而大于 FPGA 方法。当溢出数据相关系数为 0.7 时，4 种方法下的系统转存性能与溢出数据相关系数为 0.5 时基本相同。由此可见，随着溢出数据量的增加，所有系统的转存能耗均有不同程度的递增。对比 4 种控制系统，本文中方法下的缓存数据溢出控制系统性能优于其他几种方法。

（下转第 65 页）