图形处理器状态管理与故障管理方法研究

刘 晖[1,2],田 泽[1,2],聂 婴[1,2],张宏伟[3]
(1. 中国航空工业西安航空计算技术研究所,陕西 西安 710119;
2. 集成电路与微系统设计航空科技重点实验室,陕西 西安 710119;
3. 西安翔腾微电子科技有限公司,陕西 西安 710119)

摘 要:图形处理器用于绘制三维图形、模拟生物环境、进行通用计算、海量数据并行处理等应用场景,是涉及多学科交叉,具有高复杂度、高实时性、高精度计算的专用处理器。在海量信息的复杂处理过程中,图形处理器的计算过程经过复杂的状态跳变输出最终结果。如何在复杂的处理过程中保障图形处理器高效、有序的运行,在故障发生时进行快速的故障分析与检测,合理的故障处理与故障隔离,保证核心功能,最小系统在异常状态下正常运行,是图形处理器应用、管理的核心问题。文中在分析图形处理运行原理的基础上,提出了图形处理状态管理、性能监控及故障定义与处理方法,规范了图形处理器驱动软件的开发,准确定位性能瓶颈,快速分析故障与处理,在最小的资源集合下保证图形处理器核心功能的正常运行。

关键词:图形处理器;状态管理;性能管理;故障定义;故障分析
中图分类号:TP31 **文献标识码**:A **文章编号**:1673-629X(2020)08-0120-04
doi:10.3969/j.issn.1673-629X.2020.08.020

Research on State Management and Fault Management of Graphics Processors

LIU Hui[1,2],TIAN Ze[1,2],NIE Zhao[1,2],ZHANG Hong-wei[3]
(1. Aeronautical Computing Technique Research Institute,Xi'an 710119,China;
2. Aeronautics Science and Technology Key Laboratory of Integrate Circuit and Micro-system Design,
Xi'an 710119,China;
3. Xiang Teng Micro-Electronics Technology Co.,Ltd.,Xi'an 710119,China)

Abstract:The graphics processor,which is a special processor involving interdisciplinary with high complexity,high real-time,high precision computing,is used in three-dimensional graphics drawing,biological environment simulation,general computing,massive data parallel processing and other application scenarios. In the complex processing of massive information,the graphics processor outputs the final result through complex state hopping. How to ensure the efficient and orderly operation of the graphics processor in the complex processing,carry out rapid fault analysis and detection when the fault occurs,reasonably deal with the fault and isolate the fault,ensure the core functions and the normal operation of the minimum system under abnormal state are the core problems of the application and management of the graphics processor. Based on the analysis of the principle of graphics processing operation,we put forward the state management,performance monitoring and fault definition and processing methods of graphic processing,standardizes the development of the graphics processor driver software,locates performance bottleneck accurately,and analyzes fault and handles quickly,ensuring the normal operation of the graphics processor core functionality in the smallest resource collection.

Key words:GPU;state management;performance management;fault definition;fault analysis

0 引 言

图形处理器经过二十多年的发展,架构经历了固定管线、分离架构可编程染色器、统一架构染色阵列的发展[1],随着图形处理功能、性能的不断提高,电路设计复杂度直线上升,内部状态寄存器容量不断增加。例如 NVIDIA 公司于 2010 年推出的 Fermi 架构

收稿日期:2019-08-25 **修回日期**:2019-12-26
基金项目:国家"十三五"预研基金项目(31513010202);核高基重大专项(2016ZX01012101-004)
作者简介:刘 晖(1986-),男,硕士,工程师,研究方向为 Soc 设计与验证;田 泽,博士,研究员,中国航空工业集团首席技术专家,研究方向为 SoC 设计方法学、嵌入式系统设计、航空专用集成电路设计等。

GTX480 图形处理器,采用 40 nm 制造工艺,片上集成 30 亿个晶体管,双精度浮点计算性能达到 768GFLOPS,内部包含了 32 678 个 32 位寄存器[2]。复杂的设计对图形处理器的运行状态管理、调试手段、性能监测与分析、极端场景使用方式等提出了新的挑战。

1　图形处理器运行原理分析

图形处理器运行基本原理如图 1 所示。图形应用程序通过主机接口发送到命令处理器单元,命令处理器进行指令编码识别、解析及预处理,并将指令编码发送到 3D 处理引擎,经过各 3D 功能单元的处理,将绘制的像素信息存储在帧缓存中,并由显示控制模块输出到显示器上。

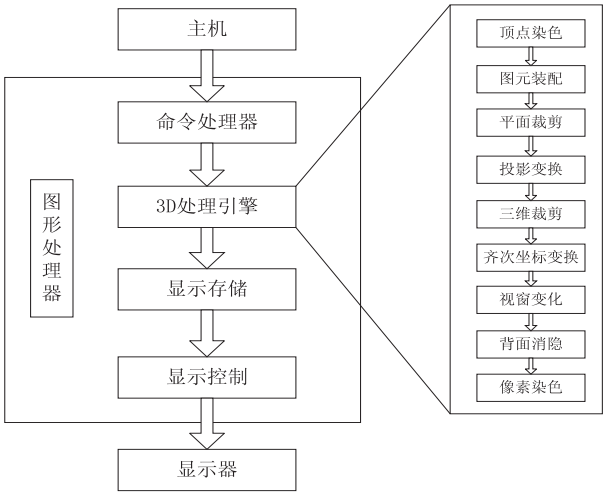

图 1　图形处理器运行原理

无论采用哪种架构实现的图形处理器,其图形处理过程与图 1 的 3D 处理引擎流程相似,不同的是,在大规模统一染色阵列架构下,任务调度单元将顶点任务与像素任务分配给不同的染色阵列核心进行计算[3]。

2　图形处理器状态管理

图形处理器运行状态管理包括了图形处理器功能运行状态管理、图形处理器性能实时管理,覆盖了图形处理器全生命周期的状态。通过状态管理可以准确地掌握图形处理的运行状态,查询相关故障,分析性能瓶颈等。

2.1　图形处理器功能状态管理

图形处理器功能运行状态如图 2 所示,包括了关闭状态、上电状态、打开状态、自检状态、初始化状态、图形绘制状态、错误状态、复位状态。各状态之间的转换须符合图 2。

(1)上电状态表示当前图形处理器处于运行准备状态,在此过程中需要检测主机端的运行空间。

(2)打开状态标志当前图形处理器处于可运行状态,在此过程中需要检测设备名称、配置运行空间。在运行空间满足图形处理器最小运行空间要求的基础上,用户可根据具体空间大小选择需要配置的运行句柄管理[4],包括设备信息管理、命令存储空间管理、GL 句柄管理、GLU 句柄管理、GLUT 句柄管理、窗口句柄管理等。

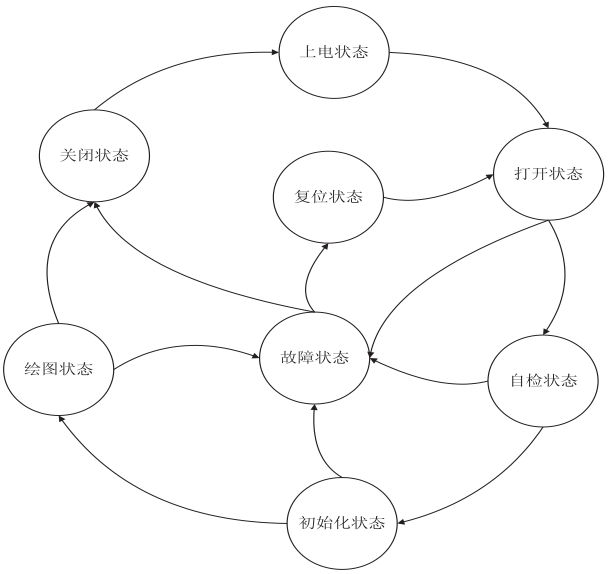

图 2　图形处理器状态

(3)自检状态用于检测图形处理器中的主要功能单元是否能够正常运行[5],包括了图形处理过程的关键路径:主机接口、3D 绘制单元、显示存储、显示控制。

(4)初始化状态标志当前图形处理器已达到运行的初始状态,在此过程中需要配置图形处理各功能单元的寄存器及存储器状态,使其达到可运行状态。

(5)绘图状态标志当前图形处理器 3D 引擎处理工作状态,绘制的图形数据存储在帧缓冲区,并最终显示出来。针对 3D 处理引擎中流水的各功能单元,统计各单元的状态信息,查看是否死锁或阻塞。

(6)故障状态标志当前图形处理器运行过程中有非法状态产生,按照故障等级进行分类处理。按模块将故障分为 5 类:设备故障、GL 故障、GLU 故障、GLUT 故障、MiniGUI 故障,每一类的故障按等级分为 3 类:一般故障、非紧急故障、紧急故障。一般故障不会影响图形处理器的正常运行,对其只做故障错误记录,不进行处理;非紧急故障是指在运行过程中出现的可承受故障,虽然此类故障影响了部分功能,但仍满足运行的最小集;紧急故障属于严重故障,导致图形处理器无法运行。

(7)复位状态是图形处理器绘图完成或出错后可选择进入的状态,它将图形处理器恢复到初始化后的状态。

(8)关闭状态表示图形处理器处于下电关闭状

态,当前图形处理器不运行。

图形处理器的状态跳转覆盖软件及硬件运行过程的全生命周期,每一种状态的转换必须符合状态迁移的前提条件,便于运行流程管理、状态监控,减少故障隐患。

2.2 图形处理器性能管理

图形处理器性能管理包括了图形命令生成速率、图形命令传输速率、命令处理器解析分发速率、顶点处理速率、图元处理速率、光栅化速率、像素处理速率、帧缓存数据刷新速率、帧缓存数据显示速率[6],如图 3 所示。图形处理器按照流水线形式逐级处理各单元的数据,每一级的数据输出都为下一级的数据输入,因此在典型场景下的性能只是实时性能统计信息,并不能作为分析图形处理器性能瓶颈的依据。

图形处理器的典型性能指标包括顶点处理速率、光栅化速率、像素处理速率[7],但其都受限于图形命令生成速率、图形命令传输速率、命令处理器解析分发速率。因此在极限性能测试时应在图形命令满带宽条件下,测试不同图元在不同处理通路下的处理速率。

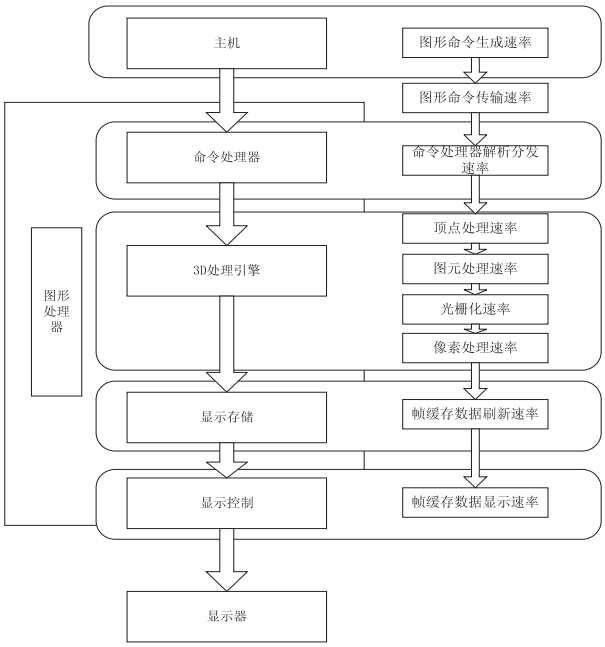

图 3 图形处理器性能分布

3 图形处理故障管理

图形处理器在运行过程中产生的故障按设计的复杂性成线性增长[8],文中只涉及图形处理模块功能的故障定义、分析及处理,不包括功能实现过程中的故障记录。由于图形处理器设计的复杂性,其故障类型千差万别,不同类型的故障不能一概而论,因此需要对故障进行分等级处理。

3.1 图形处理故障等级划分

图形处理故障按照其对整个系统的影响程度进行划分,因此识别图形处理系统的关键路径是故障等级划分的前提。

凡是影响关键路径数据处理的故障都为紧急故障[9],包括关键模块初始化状态、空间不满足系统运行的最小要求、关键计算单元自检错误等;非关键路上的故障错误按照其对绘图结果的影响和其是否可恢复分为非紧急故障和一般故障[10],非紧急故障是指软件句柄错误、软件记录错误或通过软件容错可恢复的错误等;一般故障是指图形处理标准接口定义的故障,此类故障属于标准接口故障,其处理结果与标准平台保持一致。

3.2 图形处理故障定义

图形处理器按照功能单元定义故障,如表 1 所示,故障代码按照功能模块分类。

3.3 图形处理故障分析与处理

故障代码按照类型可分为单元初始化故障、空间分配故障、句柄故障、功能单元超时故障、应用程序故障。

(1)功能单元故障影响图形处理执行流程,属于紧急故障,包括了主机接口、显示存储及显示控制[11]。主机接口是图形命令、像素数据传输的关键通路;显示存储区域可分为像素存储区,像素相关信息(如深度信息、模板信息等)存储区、视频存储区,若与当前应用匹配,则为紧急故障,否则为非紧急故障;显示控制通路可分为图像通路、视频通路,若与当前应用匹配,则为紧急故障,否则为非紧急故障。

(2)空间分配故障包括了句柄空间分配故障与运行空间分配故障,句柄空间用来记录图形处理状态,属于非关键空间,即为非紧急故障;运行空间可分为设备运行空间与最小运行空间,设备运行空间为用户设定的图形处理能够流畅处理的空间要求[12],最小空间为图形处理运行要求的最小空间,若小于该空间,则图形处理器无法运行,因此运行空间分配故障按照匹配关系可分为非紧急故障与紧急故障。

(3)句柄故障包括了设备句柄故障、GL 句柄故障、GLU 句柄故障、GLUT 句柄故障、MiniGUI 句柄故障,属于非紧急故障

(4)功能单元超时故障包括命令缓冲区超时和 3D 引擎功能单元超时。功能单元超时后可通过复位恢复正常状态,对于小概率的此类故障,且不影响整体功能时,属于非紧急故障,否则属于紧急故障[13]。

(5)应用程序故障包括非法参数、非法枚举值、非法操作、堆栈溢出、空间越界等,他们记录的是图形命令接口参数错误、调用错误,这些错误在驱动软件层做了规避操作,不会导致访问异常地址,除 0 等非法错误,属于一般故障[14]。

表 1　图形处理器故障定义

功能单元	故障代码	故障意义	故障等级
主机接口	GPU_PCI_INIT_FAILURE	主机接口初始化失败	紧急故障
	GPU_HANDLER_ERROR	设备句柄错误	非紧急故障
	GPU_ALLOC _ERROR	主机空间不满足要求	非紧急故障
	GPU_HANDLER_SELECT	有选择性的记录句柄信息	非紧急故障
命令缓冲	GPU_RING_BUFFER_INIT_FAILURE	命令缓冲区初始化失败	紧急故障
	GPU_RUNBUFFER_ALLOC_ERROR	命令缓冲空间不满足要求	紧急故障
	GPU_RINGBUFFER_TIMEOUT_ERROR	命令缓冲操作超时	非紧急故障
3D 引擎	GPU_GL_INIT_FAILURE	GL 空间初始化失败	非紧急故障
	GPU_GL_HANDLER_ERROR	GL 句柄错误	非紧急故障
	GPU_3D_FUNCTIONi_TIMEOUT_ERROR	功能单元 i 操作超时	非紧急故障
	GPU_SHADER_LOAD_ERROR	染色器程序加载错误	紧急故障
	GL_INVALID_ENUM	非法枚举值	一般故障
	GL_INVALID_VALUE	非法参数	一般故障
	GL_INVALID_OPERATION	非法操作	一般故障
	GL_STACK_OVERFLOW	堆栈上溢	一般故障
	GL_STACK_UNDERFLOW	堆栈下溢	一般故障
	GL_OUT_OF_MEMORY	超出 GPU 空间范围	一般故障
显示存储	GPU_MEMORY_INIT_FAILURE	显示存储初始化错误	紧急故障
	GPU_MEMORY_ACCESS_ERROR	存储空间访问错误	紧急故障
显示控制	GPU_DISPLAY_INIT_FAILURE	显示控制初始化错误	紧急故障

4　结束语

图形处理器以其优越的图形处理能力,具有广泛的应用前景。文中介绍了一种通用的图形处理器状态管理及性能监测方法,能够严格控制状态间的转换,监控图形处理系统性能[15],在保证图形处理器正常运行的前提下,提出了一种分级的故障管理方法,保证在最小资源集合下满足不同的应用场景。

参考文献:

[1] 王利祥. 嵌入式图形处理器的研究与设计[J]. 电子制作,2018(14):7-8.

[2] 焦继业. 低功耗高性能移动图形顶点处理器设计关键技术研究[D]. 西安:西安电子科技大学,2013.

[3] 王海峰. 图形处理器通用计算的功耗分析与优化研究[D]. 上海:上海理工大学,2013.

[4] 李　靖. 嵌入式计算机智能图像信息处理系统设计与实现[J]. 数字技术与应用,2018,36(9):158-159.

[5] 张竞旭. 基于 AMD Tonga 架构的 GPU 性能预测研究[D]. 西安:西安电子科技大学,2018.

[6] 张　军. 基于线程调度的通用图形处理器性能优化方法研究[D]. 武汉:武汉大学,2018.

[7] 唐　滔,彭　林,黄　春,等. 面向存储层次设计优化的 GPU 程序性能分析[J]. 计算机科学,2017,44(12):1-10.

[8] JIN Xingxing, DAKU B, KO S B. Improved GPU SIMD control flow efficiency via hybrid warp size mechanism[J]. Microprocessors and Microsystems,2014,38(7):717-729.

[9] YU Licheng, TANG Xingsheng, WU Minghui, et al. Improving branch divergence performance on GPGPU with a new PDOM stack and multi-level warp scheduling[J]. Journal of Systems Architecture,2014,60(5):420-430.

[10] 田　泽,张　骏,许宏杰,等. 图形处理器低功耗设计技术研究[J]. 计算机科学,2013,40(6A):210-216.

[11] 林一松. 面向 GPU 的低功耗软件优化关键技术研究[D]. 长沙:国防科学技术大学,2012.

[12] 吕相文. 高性能计算云环境下 GPU 并行计算技术及应用研究[D]. 南京:南京航空航天大学,2015.

[13] 任向隆,田　泽,张　骏,等. Catmull-Rom 图像缩放算法的自适应结构设计与实现[J]. 微电子学与计算机,2019,36(8):39-44.

[14] 张　骏,田　泽,郭　亮,等. 面向 GPU 统一染色阵列的并行自适应看门狗[J]. 航空计算技术,2018,48(5):187-193.

[15] 马城城,田　泽,黎小玉,等. 统一渲染架构 GPU 图形处理量化性能模型研究[J]. 电子技术应用,2019,45(2):27-32.