



ORIGINATE DATE
17 January, 2002

EDIT DATE
9 October, 2015

DOCUMENT-REV. NUM.
GEN-CXXXXX-REVA

PAGE
1 of 43

Author: Andrew Gruber, Andi Skende

Issue To:

Copy No:

Shader Processor

Rev 1.2

Overview: This document describes the overall architecture of the Shaders, interfaces, partitioning into functional blocks as well as the timing of the shader pipeline. It's intended for use by hardware designers.

AUTOMATICALLY UPDATED FIELDS:

Document Location : //ma_andi_mobile/.../doc_lib/parts/sp

Current Intranet Search Title: Shader Processor

APPROVALS

Name/Dept	Signature/Date

Remarks

THIS DOCUMENT CONTAINS [REDACTED] INFORMATION THAT COULD BE SUBSTANTIALLY DETRIMENTAL TO THE INTEREST OF ATI TECHNOLOGIES INC. THROUGH UNAUTHORIZED USE OR DISCLOSURE.

"Copyright 2001, ATI Technologies Inc. All rights reserved. The material in this document constitutes an unpublished work created in 2001. The use of this copyright notice is intended to provide notice that ATI owns a copyright in this unpublished work. The copyright notice is not an admission that publication has occurred. This work contains [REDACTED] proprietary information and trade secrets of ATI. No part of this document may be used, reproduced, or transmitted in any form or by any means without the prior written permission of ATI Technologies Inc."



ORIGINATE DATE
17 January, 2002

EDIT DATE
9 October, 2015

DOCUMENT-REV. NUM.
R400 Shader Processor Model

PAGE
2 of 43

Table Of Contents

STATE **6**

1.1 Shader State 6

 1.1.1 GPRs (General Purpose Registers) 6

 1.1.2 Constant Registers 6

 1.1.3 Previous Instruction Result 7

1.2 Initial state 7

 1.2.1 Vertex Shader 7

 1.2.2 Pixel Shader 7

2. PROGRAM FORMAT **7**

3. ALU **7**

3.1 ALU structure 7

3.2 ALU instruction format 7

 3.2.1 ALU Instruction Word Interpretation 9

 3.2.1.1 Relative vs. Absolute Constants 9

 3.2.1.2 Argument Selection and Pointers 9

 3.2.1.3 Input and Output Modifiers 9

 3.2.1.4 Export and Predicate related decoding 10

 3.2.1.5 Export Types and Addresses 11

3.3 ALU Opcodes 11

3.4 DX9.0 Shader Instructions, related exceptions and corner cases 14

3.5 Macro opcodes 14

4. SHADER BLOCK DIAGRAMS **15**

4.1 Shader as an SIMD architecture 15

4.2 Top-Level Diagram of a Shader Pipeline 15

5. INTERFACES **16**

5.1 External Interfaces 16

 5.1.1 Naming Convention 16

 5.1.2 Shader Engine to Texture Fetch Unit Bus 17

 5.1.3 Sequencer to Shader Pipe(s): Texture stall 17

 5.1.4 ScanConverter to Shader Pipe: IJ bus 17

 5.1.5 Sequencer to Shader Pipe(s) - broadcast: Interpolator bus 18

 5.1.6 Sequencer to Shader Pipe(s)-broadcast: Parameter Cache Read control bus 18

 5.1.7 Sequencer to Shader Pipe: GPR, Parameter Cache control and auto counter 18

 5.1.8 Shader Pipe to Shader Export (SX): Parameter data out of Parameter Cache 19

 5.1.9 Shader Export (SX) to Interpolators: Parameter Cache Return bus 19

 5.1.10 Shader Pipe to Shader Export (SX): Pixel/Vertex write to SX 19

 5.1.11 Sequencer to SPx: Instruction Interface 20

 5.1.12 Shader Pipe to Sequencer: Constant address load 22



ORIGINATE DATE
17 January, 2002

EDIT DATE
9 October, 2015

DOCUMENT-REV. NUM.
GEN-CXXXXX-REVA

PAGE
3 of 43

5.1.13 Sequencer to SPx: constant broadcast 22

6. PARAMETER INTERPOLATION 23

7. SHADER LIMITATIONS 24

8. HARDWARE IMPLEMENTATION SPECIFICS 25

8.1 General Information on the Shader Floating Point arithmetic..... 25

8.2 Interpolators and IJ/XY Buffers 25

8.2.1 Interpolators 26

8.2.1.1 Interpolation Units 26

8.2.1.2 Parameter Selection Unit 27

8.2.1.3 Parameter Difference & Cylindrical Wrap Engine..... 27

8.2.2 GPR Write Path..... 28

8.3 Vector Unit 29

8.3.1 Vector Unit Pipeline 29

8.3.2 Argument Selection and Routing..... 30

8.3.3 Parameter Data Path..... 31

8.4 Scalar Unit..... 33

8.4.1 Scalar Engine Pipeline 33

8.4.1.1.1.1 High Precision Pipeline Exp₂ Preprocessing..... 34

8.4.1.1.1.2 High Precision Pipeline Mantissa Calculation..... 34

8.4.1.1.1.3 High Precision Pipeline Log₂ Post Processing..... 38

8.4.1.1.1.4 High Precision Pipeline Exponents 38

8.4.1.1.1.5 High Precision Special Outputs..... 39

8.4.1.1.1.6 Determination of High Precision Coefficients..... 39

9. OPEN ISSUES 42



ORIGINATE DATE
17 January, 2002

EDIT DATE
9 October, 2015

DOCUMENT-REV. NUM.
R400 Shader Processor Model

PAGE
4 of 43

Revision Changes:

Rev 0.0 (Steve Morein)

Date: April, 2001
Initial revision.

Document started

Rev 0.1 (Andi Skende)

Date: May 09, 2001

Updated, added the instruction format, initial block diagrams and preliminary interface description

Rev 0.2 (Andi Skende)

Date: May 21, 2001

Rev 0.3 (Andi Skende)

Date: June 19, 2001

A more detailed description of the SP<->TEX, RE/Sequencer <->SP interfaces.

Added the paragraph related to shader functional limitations that the compiler needs to be aware of. A new updated and compressed version of ALU instruction format.

Rev 0.4 (Andi Skende)

Date: June 20, 2001

Rev 0.5 (Andi Skende)

Date: July 31, 2001

Updated the Introduction of this document. A new Pipeline Timing Diagram was inserted.

Merged in the Shader Hardware Spec. A more detailed description of the interfaces with the other blocks was added. Updated some of the diagrams to a more correct representation of the datapaths.

Rev 0.6 (Andi Skende)

Date: August 17, 2001

A more detailed description/definition of Shader interfaces with the other blocks.

A more detailed description of the instruction supported by Shader Processor and its relation to instruction set exposed at API level.

Rev 0.7 (Andi Skende)

Date: November 8, 2001

Updated the ALU instruction word definition and the list of the ALU instruction opcodes supported by the shader pipe ALU unit.

Rev 0.8 (Andi Skende)

Date: November 27, 2001

Rev 0.9 (Andi Skende)

Date: December 10, 2001

Updated the definition of the External Interfaces

Updated the definition and naming of some of the external interfaces, rearranged the ALU instruction word definition such that the fields are dword aligned. The instruction opcode definition was updated and expanded.

Rev 1.0 (Andi Skende)

Date: January 15, 2002

Updated most of the diagrams. Updated the External Interface definitions. Added a description of the Parameter Interpolation Units. Added a diagram description of the GPR write data paths.

Rev 1.1 (Andi Skende)

Date: January 21, 2002

Updated some of the external interface definitions. Specified the expected behavior of hardware implementation of some shader opcode with some corner case values as input arguments. The MS Reference Rasterizer shader was used as guideline.

Rev 1.2 (Andi Skende)

Date: January 22, 2002

Updated some of the external interface definitions.



ORIGINATE DATE
17 January, 2002

EDIT DATE
9 October, 2015

DOCUMENT-REV. NUM.
GEN-CXXXXX-REVA

PAGE
5 of 43

Introduction

Shader Pipe (SP) serves as the central Arithmetic and Logic Unit (ALU) for the R400 Graphics Processor. There are four identical Shader pipelines in the R400 architecture. Differently from previous ATI architectures, the R400 Shader Pipe truly represents an Unified Shader Architecture. In R400, both vertex and pixel shading operations are implemented through the shader units. The R400 Shader Pipe represents an SIMD architecture. All the shader units of each and every pipe execute the same ALU instruction on different sets of vertex parameters/pixel values. The building blocks of the R400 shader units execute operations on single precision IEEE floating-point values.

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.