**Goblin-Core Architecture Specification**

Version 1.0.0

Legal Information

**Change Log**

|  |  |
| --- | --- |
| Date | Description |
| 06/26/13 | v.1.0.0 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Table of Contents**

1. Introduction 5

1.1. Revision 5

1.2. Authors 5

1.3. Terms and Abbreviations 5

1.4. References 5

2. Architecture Overview 5

3. Addressing Modes & Operand Conventions 5

3.1. Addressing Modes 6

3.1.1. Local Register Indirect with Displacement 6

3.1.2. Global Register Indirect with Displacement 6

3.1.3. Instruction Pointer [IP] Relative 6

3.2. Operand Conventions 6

4. Execution Modes 7

4.1. Scalar Mode 7

4.2. Vector Mode 7

5. Exception Model 7

6. Memory Management Units 7

6.1. Local MMU 7

6.2. Global MMU 7

7. Hazard Model 7

8. Interrupt Model 7

9. Instruction Set 7

10. Debug Model 8

11. Application Binary Interface 8

11.1. Function Call Conventions 8

11.1.1. Call Semantics 8

11.1.2. Parameter Lists 8

11.1.3. Return Codes 8

11.2. System Calls 8

12. Register Organization 8

12.1. Scalar Register Indexing 8

12.2. Vector Register Indexing 10

12.3. GCONST Register 12

12.4. Performance Counter Access 12

13. Instruction Set Architecture 12

13.1. Instruction Set Format 12

13.2. Instruction Descriptions 12

14. Appendix A. Instruction Set Table 12

15. Appendix B. Performance Counters 12

16. Appendix NNNN. 12

**Table of Figures**

# Introduction

# Revision

**Version 1.0.0**: Initial Release of Goblin-Core 64 Architecture Specification

# Authors

**Architecture Lead:** John D. Leidel, Texas Tech University

# Terms and Abbreviations

**ABI** – Application Binary Interface specification.

**GC64** – Abbreviation for the Goblin-Core 64-bit processor architecture.

**Rn** – General purpose register [where ‘n’ is 0-31]

**Vn** – Vector register [where ‘n’ is 0-7]

# References

# Architecture Overview

# Addressing Modes & Operand Conventions

The following section describes the local and global address mode conventions for the Goblin-Core architecture family. This includes local and global address modes as well as the associated operand conventions as they appear in the instruction set mnemonics.

# Addressing Modes

# Local Register Indirect with Displacement

# Global Register Indirect with Displacement

# Instruction Pointer [IP] Relative

# Operand Conventions

The architecture defines 8-bit byte word, 16-bit word, 32-bit double word, and 64-bit quad word. The architecture defines an IEEE-754 32-bit single precision floating point representation as well as an IEEE-754 64-bit double precision floating point representation. The architecture also defines a 128-bit global addressing format. The most significant quad word [64-bits] contains the address portion of the global address operand. The least significant quad word [64-bits] contains the interconnect locality portion of the address operand. However, 128-bit global address operands map to 64-bit quad word data payloads.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Mnemonic | Size [bytes] | Notes |
| Unsigned Byte | UB | 1 |  |
| Unsigned Word | UW | 2 |  |
| Unsigned Double | UD | 4 |  |
| Unsigned Quad | UQ | 8 |  |
| Signed Byte | SB | 1 |  |
| Signed Word | SW | 2 |  |
| Signed Double | SD | 4 |  |
| Float Single | FS | 4 | IEEE-754 32-bit single precision floating point |
| Float Double | FD | 8 | IEEE-754 64-bit double precision floating point |
| Global Address | GA | 16 | Global Addressing operand mode. Most significant quad contains the address portion of the operand. Least significant quad contains the interconnect locality portion. |

**Figure 1. Operand Conventions**

# Execution Modes

# Scalar Mode

# Vector Mode

# Exception Model

# Memory Management Units

# Local MMU

# Global MMU

# Hazard Model

# Interrupt Model

# Instruction Set

# Debug Model

# Application Binary Interface

# Function Call Conventions

# Call Semantics

# Parameter Lists

# Return Codes

# System Calls

# Register Organization

The GC64 architecture defines a set of user-accessible user registers. These user registers are split into two banks, user writable registers and architecture constants. Registers are accessed via their direct index values as register renaming is not supported in the GC64 architecture. As such, the maximum number of user registers per the instruction set format is 64 [index 0x3F]. All read+write registers are indexed 0x00 – 0x34. All read-only registers are indexed 0x35 - 0x3F.

# Scalar Register Indexing

The GC64 base scalar mode contains thirty two [32] general-purpose scalar user registers. These registers are referred to as ‘R’ registers in the assembly mnemonics [R0-R31]. The remainder of the read-write registers have a pre-defined purpose, respectively. The following table defines the set of scalar-accessible registers.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Index | Name | Size [bits] | Mode | Description |
| 0x00 | R0 | 64 | Read-Write | General register |
| 0x01 | R1 | 64 | Read-Write | General register |
| 0x02 | R2 | 64 | Read-Write | General register |
| 0x03 | R3 | 64 | Read-Write | General register |
| 0x04 | R4 | 64 | Read-Write | General register |
| 0x05 | R5 | 64 | Read-Write | General register |
| 0x06 | R6 | 64 | Read-Write | General register |
| 0x07 | R7 | 64 | Read-Write | General register |
| 0x08 | R8 | 64 | Read-Write | General register |
| 0x09 | R9 | 64 | Read-Write | General register |
| 0x0A | R10 | 64 | Read-Write | General register |
| 0x0B | R11 | 64 | Read-Write | General register |
| 0x0C | R12 | 64 | Read-Write | General register |
| 0x0D | R13 | 64 | Read-Write | General register |
| 0x0E | R14 | 64 | Read-Write | General register |
| 0x0F | R15 | 64 | Read-Write | General register |
| 0x10 | R16 | 64 | Read-Write | General register |
| 0x11 | R17 | 64 | Read-Write | General register |
| 0x12 | R18 | 64 | Read-Write | General register |
| 0x13 | R19 | 64 | Read-Write | General register |
| 0x14 | R20 | 64 | Read-Write | General register |
| 0x15 | R21 | 64 | Read-Write | General register |
| 0x16 | R22 | 64 | Read-Write | General register |
| 0x17 | R23 | 64 | Read-Write | General register |
| 0x18 | R24 | 64 | Read-Write | General register |
| 0x19 | R25 | 64 | Read-Write | General register |
| 0x1A | R26 | 64 | Read-Write | General register |
| 0x1B | R27 | 64 | Read-Write | General register |
| 0x1C | R28 | 64 | Read-Write | General register |
| 0x1D | R29 | 64 | Read-Write | General register |
| 0x1E | R30 | 64 | Read-Write | General register |
| 0x1F | R31 | 64 | Read-Write | General register |
| 0x20 | SP | 64 | Read-Write | Stack Pointer |
| 0x21 | FP | 64 | Read-Write | Frame Pointer |
| 0x22 | PIC | 64 | Read-Write | Processor Independent Code Pointer |
| 0x23 | UP | 64 | Read-Write | Uplevel Frame Pointer |
| 0x24 | RP | 64 | Read-Write | Return Pointer |
| 0x25 | CC0 | 64 | Read-Write | Compare Register 0 |
| 0x26 | CC1 | 64 | Read-Write | Compare Register 1 |
| 0x27 | CC2 | 64 | Read-Write | Compare Register 2 |
| 0x28 | CC3 | 64 | Read-Write | Compare Register 3 |
| 0x29 | TQ | 64 | Read-Write | Task Queue |
| 0x2A | VL | 64 | Read-Write | Vector Length |
| 0x2B | VS | 64 | Read-Write | Vector Stride |
| 0x2C | GMODE | 64 | Read-Write | GC64 Mode |
| 0x2D | PMASK | 64 | Read-Write | Performance Counter Mask |
| 0x2E | PREAD | 64 | Read-Write | Performance Counter Read Index |
| 0x2F | TID | 64 | Read-Write | Thread ID |
| 0x30 | (--) | (--) | (--) | (--) |
| 0x31 | (--) | (--) | (--) | (--) |
| 0x32 | (--) | (--) | (--) | (--) |
| 0x33 | (--) | (--) | (--) | (--) |
| 0x34 | (--) | (--) | (--) | (--) |
| 0x35 | ZERO | 64 | Read-Only | Zero: 0x00LL |
| 0x36 | IMM32 | 64 | Read-Only | 4-Byte Immediate |
| 0x37 | IMM64 | 64 | Read-Only | 8-Byte Immediate |
| 0x38 | GCONST | 64 | Read-Only | GC64 Architecture Constant |
| 0x39 | EQ | 64 | Read-Only | Compare Val: Equal |
| 0x3A | GT | 64 | Read-Only | Compare Val: Greater Than |
| 0x3B | GTE | 64 | Read-Only | Compare Val: Greater Than + Equal |
| 0x3C | LT | 64 | Read-Only | Compare Val: Less Than |
| 0x3D | LTE | 64 | Read-Only | Compare Val: Less Than + Equal |
| 0x3E | NE | 64 | Read-Only | Compare Val: Not Equal |
| 0x3F | PVAL | 48 | Read-Only | Performance Counter Value |

**Figure N. Register Index Table**

# Vector Register Indexing

The GC64 architecture supports an aliased vector indexing mode into the aforementioned register file. Vector indexes map to multiple, strided scalar *%Rn* registers. Vector registers are denoted with the mnemonic *%Vn* by the assembler. Changing index modes from scalar [*%Rn*] to vector [*%Vn*] and vice versa does not modify the contents of the register. The only registers available via the vector aliased mode are the general purpose registers. All control and constant registers are always considered scalars.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Vector Register | Vector Element | Scalar Register | Register Index | Size [bits] |
| V0 | Elem 0 | R0 | 0x00 | 64 |
| V0 | Elem 1 | R1 | 0x01 | 64 |
| V0 | Elem 2 | R2 | 0x02 | 64 |
| V0 | Elem 3 | R3 | 0x03 | 64 |
| V1 | Elem 0 | R4 | 0x04 | 64 |
| V1 | Elem 1 | R5 | 0x05 | 64 |
| V1 | Elem 2 | R6 | 0x06 | 64 |
| V1 | Elem 3 | R7 | 0x07 | 64 |
| V2 | Elem 0 | R8 | 0x08 | 64 |
| V2 | Elem 1 | R9 | 0x09 | 64 |
| V2 | Elem 2 | R10 | 0x0A | 64 |
| V2 | Elem 3 | R11 | 0x0B | 64 |
| V3 | Elem 0 | R12 | 0x0C | 64 |
| V3 | Elem 1 | R13 | 0x0D | 64 |
| V3 | Elem 2 | R14 | 0x0E | 64 |
| V3 | Elem 3 | R15 | 0x0F | 64 |
| V4 | Elem 0 | R16 | 0x10 | 64 |
| V4 | Elem 1 | R17 | 0x11 | 64 |
| V4 | Elem 2 | R18 | 0x12 | 64 |
| V4 | Elem 3 | R19 | 0x13 | 64 |
| V5 | Elem 0 | R20 | 0x14 | 64 |
| V5 | Elem 1 | R21 | 0x15 | 64 |
| V5 | Elem 2 | R22 | 0x16 | 64 |
| V5 | Elem 3 | R23 | 0x17 | 64 |
| V6 | Elem 0 | R24 | 0x18 | 64 |
| V6 | Elem 1 | R25 | 0x19 | 64 |
| V6 | Elem 2 | R26 | 0x1A | 64 |
| V6 | Elem 3 | R27 | 0x1B | 64 |
| V7 | Elem 0 | R28 | 0x1C | 64 |
| V7 | Elem 1 | R29 | 0x1D | 64 |
| V7 | Elem 2 | R30 | 0x1E | 64 |
| V7 | Elem 3 | R31 | 0x1F | 64 |

**Figure N. Vector Register Aliasing Table**

# GCONST Register

# Performance Counter Access

# Instruction Set Architecture

# Instruction Set Format

# Instruction Descriptions

# Appendix A. Instruction Set Table

# Appendix B. Performance Counters

# Appendix NNNN.