Micro-Code

© 2023 Robert Finch

# Overview

Micro-code is software that the CPU uses internally to perform complex operations. It is hidden within the CPU and not generally available for programming. Qupls has micro-code to support complex functions like reciprocal square root. Micro code consists of a mix of regular instructions and micro-code specific ones.

Micro-code is not exposed as part of the ISA, instead it is implementation specific. Whether the implementation uses micro-code or a state machine it is opaque.

# Registers

To execute micro-code program effectively, temporary registers hidden from the programming model need to be available. There are five hidden architectural registers for use with micro-code. Whether a general-purpose register or micro-code use register is selected based on a flag value in the micro-code. Micro-code architectural registers share the register renaming mechanism with general purpose registers. There are enough additional physical registers to accommodate both sets of registers.

# Instruction Set

The entire Qupls instruction set is available to micro-code with a few exceptions.

## Micro-Code Branches

**Overview**

Micro-code branches are reserved for micro-code use. They share the branch opcodes of the instruction set, but the target is formed differently. The CPU can distinguish whether to use ordinary branches or micro-code branches based on whether micro-code is active.

**Instruction Format**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ▼ | ▼ |  |  |  |  |  |  |
| 39 37 | 36 25 | 24 19 | 18 13 | 12 | 11 | 10 7 | 6 0 |
| ~ | Target11..0 | Rb6 | Ra6 | ~ | Lk | Fn4 | 2xh7 |

#### **Linkage**

Branches may specify a linkage register which is updated with the address of the next instruction. This allows subroutines to be called. Only the *micro-code* link register may be updated.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | ▼ |  |  |
| 39 37 | 36 25 | 24 19 | 18 13 | 12 | 11 | 10 7 | 6 0 |
| ~ | Target11..0 | Rb6 | Ra6 | ~ | Lk | Fn4 | 2xh7 |

#### **Branch Target**

For micro-code conditional branches, the target address is an absolute micro-code address. Only the micro-code IP is updated.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ▼ | ▼ |  |  |  |  |  |  |
| 39 37 | 36 25 | 24 19 | 18 13 | 12 | 11 | 10 7 | 6 0 |
| ~ | Target11..0 | Rb6 | Ra6 | ~ | Lk | Fn4 | 2xh7 |

# Micro-coded Operations

Reset

Micro-code performs a reset routine when reset is activated. The initial stack pointer and program counter are loaded from vectors in memory.

Multi-precision Floating-Point Operations

Reciprocal Estimate

Reciprocal square root

Divide

Sine and Cosine