*Project: RMC75E FPGA TEST BENCH*

*Module:* *CPUconfig.vhd*

*Author: Satchel Hamilton*

*Company: Delta Motion*

*Date: 7/13/2023*

*Last updated: July 13, 2023*

Contents

[High Level 1](#_Toc140670735)

[Low level: 3](#_Toc140670736)

[Simulation: 4](#_Toc140670737)

# High Level

The **CPUConfig** module is responsible for handling various configurations related to the CPU and control loop parameters. It takes multiple input signals and processes them to control several output signals, including the CPU drive enable (**M\_DRV\_EN\_L**) line, the control loop time selection (**LoopTime**), and the reset signal for internal components (**DLL\_RST**). The module receives a 32-bit data input (**intDATA**) and a control signal (**CPUConfigWrite**) to update its internal configuration registers. It also receives reset signals (**RESET** and **SysRESET**) and a clock signal (**H1\_CLKWR** and **H1\_PRIMARY**) for synchronization. The module also interfaces with the HALT drive signal (**HALT\_DRIVE\_L**) and the Ethernet build signal (**ENET\_Build**).

# Low level

The **CPUConfig** module is implemented in the **CPUConfig\_arch** architecture. It consists of signal declarations and two process blocks that handle the update of internal signals based on the input conditions.

Signal Declarations:

* **int\_M\_DRV\_EN**: A signal used to handle the drive enable line (**M\_DRV\_EN\_L**). The actual drive enable line is active low, but the processor writes to this signal as if it's active high.
* **int\_DLL\_RST**: A signal used to handle the DLL reset (**DLL\_RST**) condition.
* **intLoopTime**: A 3-bit signal used for control loop time selection.
* **dll\_rst\_pre\_queue**: A signal used for queueing the DLL reset signal to avoid glitches.
* **dll\_rst\_queue**: A 2-bit signal used for queueing the DLL reset to avoid glitches.

Output Signals:

* **cpuConfigDataOut**: The output signal concatenates several values, including the control loop time selection, DLL lock status, and drive enable status, to communicate with external components.
* **M\_DRV\_EN\_L**: The drive enable output signal, active low, controls the drives.
* **DLL\_RST**: The output signal is used to clear the SysRESET condition.

Process Blocks:

1. The first process block handles the CPU drive enable (**M\_DRV\_EN\_L**) line and the control loop time selection (**intLoopTime**). It sets **int\_M\_DRV\_EN** based on **HALT\_DRIVE\_L** and updates the **intLoopTime** value when **CPUConfigWrite** is active and there is a rising edge on **H1\_CLKWR**. The **intLoopTime** is updated with the three least significant bits of **intDATA**, and **M\_DRV\_EN\_L** is driven with the inverted value of **int\_M\_DRV\_EN**.
2. The second process block handles the DLL reset (**DLL\_RST**). It clears the **int\_DLL\_RST** signal when **SysRESET** is low and updates it based on **intDATA(2)** when **CPUConfigWrite** is active and there is a rising edge on **H1\_PRIMARY**. To prevent glitches, the **dll\_rst\_pre\_queue** is used to queue the DLL reset signal when no writes are occurring (**CPUConfigWrite = '0'**), and **dll\_rst\_queue** holds the previous and current value to generate the **DLL\_RST** signal.

## Simulation

To verify the functionality of the **CPUConfig** module, simulation tests can be performed. In the simulation, appropriate test vectors for the input signals (**RESET**, **SysRESET**, **H1\_CLKWR**, **H1\_PRIMARY**, **intDATA**, **CPUConfigWrite**, **HALT\_DRIVE\_L**, and **ENET\_Build**) are provided. Observing the behavior of the output signals (**M\_DRV\_EN\_L**, **LoopTime**, and **DLL\_RST**) over time will show how the module responds to changes in inputs and how it handles the control loop time and reset conditions.