*Project: RMC75E FPGA TEST BENCH*

*Module: module.vhd*

*Author: Satchel Hamilton*

*Company: Delta Motion*

*Date: 7/13/2023*

*Last updated: July 13, 2023*

Contents

[High Level 1](#_Toc141279393)

[Low level 1](#_Toc141279394)

[Simulation 2](#_Toc141279395)

# High Level

**Description:** The entity "DiscoverExpansionID" represents a module in the source code for the RMC75E modular motion controller. This VHDL module is responsible for discovering the ID information from the expansion modules after a reset.

# Low level

**Design Details:** The DiscoverExpansionID module is designed to query ID information from expansion modules. It uses a state machine and a shift register to capture the ID data provided by an external module through Exp\_ID\_DATA. The module generates an output clock signal (Exp\_ID\_CLK) and controls the signals Exp\_ID\_LOAD and Exp\_ID\_LATCH to facilitate the data capturing process. The captured ID information is stored in four separate 17-bit signals (ExpansionID0, ExpansionID1, ExpansionID2, ExpansionID3).

**Inputs:**

* RESET: A synchronous reset signal to initialize the state machine and shift register.
* SysClk: The system clock signal used for synchronous operations.
* SlowEnable: A control signal that enables the generation of a 3.75MHz clock output for the serial communication state machine.
* Exp\_ID\_DATA: The data input signal containing the ID information from an external module.

**Outputs:**

* ExpansionID0: A 17-bit bidirectional signal that holds the ID information from Expansion Module 0.
* ExpansionID1: A 17-bit bidirectional signal that holds the ID information from Expansion Module 1.
* ExpansionID2: A 17-bit bidirectional signal that holds the ID information from Expansion Module 2.
* ExpansionID3: A 17-bit bidirectional signal that holds the ID information from Expansion Module 3.
* Exp\_ID\_CLK: The output clock signal for the state machine.
* Exp\_ID\_LATCH: The output signal to control the latch signal for capturing the ID data.
* Exp\_ID\_LOAD: The output signal to control the load signal for capturing the ID data.

**Architecture:** The architecture of the DiscoverExpansionID module, named DiscoverExpansionID\_arch, consists of two processes.

1. StateMachine process: This process controls the state transitions based on the system clock (SysClk) and the SlowEnable signal. It drives the control signals Exp\_ID\_LOAD, Exp\_ID\_LATCH, and ShiftEnable. The state machine goes through different states to perform the capture operation. It starts with s0\_LatchState, proceeds to s1\_DelayState1, s2\_LoadState, s3\_DelayState2, s4\_ClockState, and finally reaches s5\_StopState.
2. Process for Shift Register and Counters: This process handles the data capture and shifting process. It captures the Exp\_ID\_DATA and performs a shift operation on the ExpansionID signals based on the OutputClock signal generated by the state machine. It also handles the Count signal, which serves as a 6-bit synchronous counter with count enable. The process uses TerminalCountValue ("111111") to determine the terminal count value, and when the count reaches this terminal count and SlowEnable is '1', the ShiftComplete signal is set to '1'. ShiftComplete is used to indicate the completion of the shift operation.
3. Clock Generation:
   * The module receives the system clock **SysClk**, which is used to synchronize internal operations.
   * The **SlowEnable** signal is used to enable a 3.75MHz clock output for the serial communication state machine.
4. State Encoding:
   * The module uses a 3-bit state encoding (**STATE\_TYPE**) to represent different states in the state machine.
   * Constants like **s0\_LatchState**, **s1\_DelayState1**, etc., define the different states and their values.
5. State Machine:
   * The module employs a synchronous state machine (**StateMachine**) to manage the process of querying ID information from expansion modules.
   * After reset (**RESET** signal asserted), the state machine initializes in **s0\_LatchState**.
6. Control Signals Initialization:
   * On reset, certain control signals (**Exp\_ID\_LOAD**, **Exp\_ID\_LATCH**, and **ShiftEnable**) are initialized to specific values.
   * **Exp\_ID\_LOAD** is set to '1', **Exp\_ID\_LATCH** is set to '0', and **ShiftEnable** is set to '0'.
7. State Transitions:
   * The state machine transitions through different states based on the current state and other conditions.
   * The **SlowEnable** signal is used to enable state transitions and control the operation of the state machine.
8. Clocking Mechanism:
   * The module generates an **Exp\_ID\_CLK** signal, which is derived from the system clock (**SysClk**) and is used for clocking data during the ID querying process.
9. 6-bit Synchronous Counter:
   * The module uses a 6-bit synchronous counter (**Count**) to keep track of clock cycles during the ID querying process.
   * The counter is reset to "000000" when in the **s0\_LatchState** or when **ShiftEnable** is '0'.
   * The counter increments on each positive edge of **SysClk** when **ShiftEnable** and **OutputClock** are '1'.
10. Shift Register:
    * The module employs a 64-bit shift register to shift in data received during the ID querying process.
    * The shift register is updated when **ShiftEnable** is '1', **OutputClock** is '0', and **SlowEnable** is '1'.
    * The data from **Exp\_ID\_DATA** is serially loaded into the shift register, shifting the existing data to the right.
11. Shift Complete Detection:
    * The module uses a **TerminalCount** value (64 + 1) and a counter (**Count**) to detect the end of the shift operation (**ShiftComplete**).
    * When the counter reaches the **TerminalCount**, **ShiftComplete** is set to '1', indicating the end of the shift operation.
12. Data Valid Flag:

* The most significant bit (bit 16) of each expansion ID (**ExpansionID0**, **ExpansionID1**, **ExpansionID2**, **ExpansionID3**) acts as a data valid flag.
* It is set to '1' when the state machine is in **s5\_StopState**, indicating that the ID data is valid and can be read.

Overall, the **DiscoverExpansionID** module employs a state machine to control the process of querying ID information from expansion modules. It utilizes a shift register and a counter to synchronize and manage data reception during the querying process. The module also provides a clock output (**Exp\_ID\_CLK**) and sets a data valid flag to indicate when the ID data is ready to be accessed.

## Simulation

To simulate the DiscoverExpansionID module, a testbench can be created with appropriate stimuli for the inputs (RESET, SysClk, SlowEnable, Exp\_ID\_DATA). The simulation should observe the behavior of the ExpansionID0, ExpansionID1, ExpansionID2, and ExpansionID3 signals, as well as the Exp\_ID\_CLK, Exp\_ID\_LATCH, and Exp\_ID\_LOAD signals. The testbench should test different scenarios, including the reset condition, state transitions, and successful data capture, to ensure the correct functionality of the module.