CAN-CTRL
CAN 2.0 & CAN-FD Bus Controller Megafunction

Implements a CAN protocol bus controller that performs serial communication according to the CAN 2.0A, 2.0B, and the Bosch CAN-FD specifications.

The CAN protocol uses a multi-master bus configuration for the transfer of frames between nodes of the network and manages error handling with no burden on the host processor. The megafunction enables the user to set up economic and reliable links between various components. It appears as a memory-mapped I/O device to the host processor, which accesses the CAN megafunction to control the transmission or reception of frames.

The CAN megafunction is easy to use and integrate featuring programmable interrupts, data and baud rates; a configurable number of independently programmable acceptance filters; and a generic processor interface or optionally an AMBA APB, or AHB-Lite interface. It implements a flexible buffering scheme, allowing fine-tuning of the megafunction size to the requirements of each specific application. The number of receive buffers is synthesis-time configurable from 2 to 31. Two types of transmit buffers are implemented: a high-priority primary transmit buffer (PTB) and a lower-priority secondary transmit buffer (STB). The PTB can store one message, while the number of included buffer slots for the STB is synthesis-time configurable 0 to 16 slots. Moreover, an optional wrapper instantiating multiple CAN controllers eases integration in cases where multiple bus-nodes need to be controlled by the same host processor.

The megafunction Implements functionality similar to the Philips SJA1000 working with its PeliCAN mode extensions, providing error analysis, diagnosis, system maintenance and optimization features. Additionally, the controller offers a CAN-FD extension to increase net data throughput. It comprises data fields up to 64 bytes, extended CRC checksum, and a higher data rate for the data phase.

The CAN megafunction is robustly verified and proven in multiple production designs.

Applications
The CAN-CTRL megafunction can be integrated in devices that use CAN or higher-layer, CAN-based communication protocols. In addition to traditional automotive applications, such devices are used in industrial (e.g. the CANopen and DeviceNet protocols), aviation (e.g. the ARINC-825 and CANaerospace protocols), marine (e.g. the NMEA 2000 protocol) and other applications.

Block Diagram
Functional Description

The CAN bus megafunction is founded on the basic CAN principle and meets all constraints of the CAN-specification 2.0B.

Several 13-byte buffers are used for buffering received or transmitted messages. The number of buffers can be selected before synthesis. Selecting a large number of buffers disables the need for real-time reaction to CAN messages for the host processor, which significantly eases software development of the system application.

The included high-priority transmit buffer can be used to transmit an important message as fast as possible, even if several lower-priority messages are pending for transmission.

The host interface contains all necessary registers for controlling and configuring the megafunction. The host is able to read and write all registers as a conventional RAM in memory mapped mode.

The interface to the host is software configurable. All events on the CAN data bus or in the CAN controller megafunction are signaled using an interrupt. Every interrupt source may be individually enabled or disabled. The CAN controller megafunction contains up to sixteen software-programmable 29-bit acceptance filters that can be used to block unwanted CAN messages, which reduces the load to the host controller.

The host controller interface is connected with the memory module by an 8-bit data and a 8-bit address bus. This enables easy interfacing to many host controller types, and therefore, quick integration with a microcontroller.

Support

The megafunction as delivered is warranted against defects for ninety days from purchase. Thirty days of phone and email technical support are included, starting with the first interaction. Additional maintenance and support options are available.

Verification

The megafunction has been rigorously verified and has been production proven multiple times.

Implementation Results

CAN-CTRL reference designs have been evaluated in a variety of technologies. The following are sample results optimized for area with a clock constraint of 80 MHz, with 3 receive buffers, 3 transmit buffers and 3 acceptance filters, with and without the CAN-FD option.

<table>
<thead>
<tr>
<th>Family</th>
<th>Device</th>
<th>LEs/ALMs</th>
<th>Memory</th>
<th>Memory Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cyclone V</td>
<td>5CGXF3C-7</td>
<td>821</td>
<td>5 M10K</td>
<td>1,344</td>
</tr>
<tr>
<td>Arria V</td>
<td>5AGTM3C-5</td>
<td>822</td>
<td>5 M10K</td>
<td>1,344</td>
</tr>
<tr>
<td>Stratix V</td>
<td>5SGXMA3-2</td>
<td>819</td>
<td>5 M20K</td>
<td>1,344</td>
</tr>
<tr>
<td>Arria 10</td>
<td>10AX048-2</td>
<td>816</td>
<td>5 M20K</td>
<td>1,344</td>
</tr>
<tr>
<td>MAX 10</td>
<td>10M04-7</td>
<td>2,117</td>
<td>4 M9K</td>
<td>1,280</td>
</tr>
</tbody>
</table>

Table 1: Sample implementation data without the CAN-FD option

<table>
<thead>
<tr>
<th>Family</th>
<th>Device</th>
<th>LEs/ALMs</th>
<th>Memory</th>
<th>Memory Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cyclone V</td>
<td>5CGXF3C-7</td>
<td>1,004</td>
<td>5 M10K</td>
<td>8,960</td>
</tr>
<tr>
<td>Arria V</td>
<td>5AGTM3C-5</td>
<td>1,005</td>
<td>5 M10K</td>
<td>8,960</td>
</tr>
<tr>
<td>Stratix V</td>
<td>5SGXMA3-2</td>
<td>1,002</td>
<td>5 M20K</td>
<td>8,960</td>
</tr>
<tr>
<td>Arria 10</td>
<td>10AX048-2</td>
<td>1,002</td>
<td>5 M20K</td>
<td>8,960</td>
</tr>
<tr>
<td>MAX 10</td>
<td>10M04-7</td>
<td>2,452</td>
<td>4 M9K</td>
<td>8,960</td>
</tr>
</tbody>
</table>

Table 2: Sample implementation data with the CAN-FD option

Deliverables

The megafunction includes everything required for successful implementation:

- VHDL or Verilog RTL source code
- Post-synthesis EDIF (netlist licenses)
- Testbenches
  o Behavioral tests
  o Post-synthesis verification
- Simulation scripts
- Synthesis scripts
- Documentation