HCDC Chip Overview: configuration and control

Yipeng Huang
Columbia University
August 22, 2013
HCDC System Architecture

Desktop PC
- USB Controller

Arduino Mega 2560
- USB Controller
- ATmega2560 microcontroller
- SPI Controller

HCDC Board
- SPI Controller
- Power Supply
- HCDC Fabric
- Clock Generator

Connections:
- USB
- MOSI
- MISO
- SCLK
- SS
HCDC Chip Architecture

[Diagram showing HCDC Chip Architecture with sections labeled HCDC Tile and HCDC Fabric, featuring Functional Units, Computational Datapaths, and Control Units.]
Chip Computational Datapath
## Data Input Output Channels

<table>
<thead>
<tr>
<th>Channel Type</th>
<th>Number Available</th>
<th>Precision</th>
<th>Update Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analog simulation data input</td>
<td>4</td>
<td>Native</td>
<td>Native</td>
</tr>
<tr>
<td>Analog simulation data output</td>
<td>4</td>
<td>Native</td>
<td>Native</td>
</tr>
<tr>
<td>Digital initial condition input</td>
<td>4</td>
<td>8 bit</td>
<td>At simulation start</td>
</tr>
<tr>
<td>Digital simulation data parallel input</td>
<td>1</td>
<td>8 bit</td>
<td>As fast as ADC</td>
</tr>
<tr>
<td>Digital simulation data parallel output</td>
<td>1</td>
<td>8 bit</td>
<td>As fast as DAC</td>
</tr>
<tr>
<td>Digital simulation data serial output</td>
<td>2</td>
<td>8 bit</td>
<td>700 KHz</td>
</tr>
</tbody>
</table>
Phases of Operation

- CONFIGURATION of equation, coefficients, initial conditions
- CALIBRATION of electrical gain and offsets
- EXECUTION of circuit that simulates the desired equation
  - Indefinite operation
  - Timeout
- Gathering EXCEPTION information from completed simulation
  - Overflow (saturation)
  - Underflow (loss of precision)
Example: Modeling an Equation

\[ \ddot{\Theta} = -0.5 \dot{\Theta} - 5\sin \Theta, \quad \Theta(0) = 1 \]
Example: Mapping to Chip

\[
\int \sin(x) - 0.5x - 5x
\]

DAC

ADC

Analog Input

Analog Output
Chip Configuration Protocol

- System master provides configuration via SPI interface
- Messages are 24 bits, sent over SPI
Example: Configuration Message

000000_000001_0010_10000000

- **Tile address, can address 64 tiles (chips)**
- **Functional unit address, 000101 accesses the first integrator**
- **Line address, 0010 sets the initial condition of the integrator**
- **Value, 1000000₂ is middle value of the range we chose for the initial condition**
Example: Configuration Message

000000_000001_0101_10000000

Tile address, can address 64 tiles (chips)
Functional unit address, 000101 accesses the first integrator
Line address, 0101 sets the output destination of the integrator
Value, setting the first bit sends the output to the fanout unit

∫
DAC
ADC
Analog Input
Analog Output

\[
\int \frac{x}{2} \sin(x) - \frac{5x}{2} - 5x
\]
Example: Configuration Message

000000_000001_0101_00110000

- Tile address, can address 64 tiles (chips)
- Functional unit address, 0000001 accesses the tile controller
- Line address, 0101 accesses the lowest timeout setting register
- Value, 00110000₂ is 48₁₀, so simulator will run for 48 clock cycles, or 68.6 µs
## Chip Configuration Protocol

- Tables describe addresses to access:
  - Datapath switches
  - Parameters
  - Lookup table entries
  - Simulation timer

<table>
<thead>
<tr>
<th>bits [9:11]</th>
<th>000</th>
<th>001</th>
<th>010</th>
<th>011</th>
<th>100</th>
<th>101</th>
<th>110</th>
<th>111</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>Unmapped</td>
<td>Tile controller addresses and data [xx.xx, 00,000,001] (See Tile Controller Bits 12:23)</td>
<td>Unmapped</td>
<td>(0.0) FAN configuration and crossbar messages (See FAN and CB Bits [12:23])</td>
<td>(0.1) FAN configuration and crossbar messages (See FAN and CB Bits [12:23])</td>
<td>(0.2) INT configuration and crossbar messages (See INT and CB Bits [12:23])</td>
<td>(0.3) MUL configuration and crossbar messages (See MUL and CB Bits [12:23])</td>
<td>(0.4) MUL configuration and crossbar messages (See MUL and CB Bits [12:23])</td>
</tr>
<tr>
<td>001</td>
<td>(5.1) LUT configuration and crossbar messages (See LUT Bits [12:23])</td>
<td>(5.2) LUT configuration and crossbar messages (See LUT Bits [12:23])</td>
<td>Unmapped</td>
<td>(1.0) FAN configuration and crossbar messages (See FAN and CB Bits [12:23])</td>
<td>(1.1) FAN configuration and crossbar messages (See FAN and CB Bits [12:23])</td>
<td>(1.2) INT configuration and crossbar messages (See INT and CB Bits [12:23])</td>
<td>(1.3) MUL configuration and crossbar messages (See MUL and CB Bits [12:23])</td>
<td>(1.4) MUL configuration and crossbar messages (See MUL and CB Bits [12:23])</td>
</tr>
<tr>
<td>010</td>
<td>(4.1) ADC configuration and crossbar messages (See ADC Bits [12:23])</td>
<td>(4.2) ADC configuration and crossbar messages (See ADC Bits [12:23])</td>
<td>Unmapped</td>
<td>(2.0) FAN configuration and crossbar messages (See FAN and CB Bits [12:23])</td>
<td>(2.1) FAN configuration and crossbar messages (See FAN and CB Bits [12:23])</td>
<td>(2.2) INT configuration and crossbar messages (See INT and CB Bits [12:23])</td>
<td>(2.3) MUL configuration and crossbar messages (See MUL and CB Bits [12:23])</td>
<td>(2.4) MUL configuration and crossbar messages (See MUL and CB Bits [12:23])</td>
</tr>
<tr>
<td>011</td>
<td>(4.3) DAC configuration and crossbar messages (See DAC Bits [12:23])</td>
<td>(4.4) DAC configuration and crossbar messages (See DAC Bits [12:23])</td>
<td>Unmapped</td>
<td>(3.0) FAN configuration and crossbar messages (See FAN and CB Bits [12:23])</td>
<td>(3.1) FAN configuration and crossbar messages (See FAN and CB Bits [12:23])</td>
<td>(3.2) INT configuration and crossbar messages (See INT and CB Bits [12:23])</td>
<td>(3.3) MUL configuration and crossbar messages (See MUL and CB Bits [12:23])</td>
<td>(3.4) MUL configuration and crossbar messages (See MUL and CB Bits [12:23])</td>
</tr>
<tr>
<td>100</td>
<td>101</td>
<td>110</td>
<td>111</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUT 1 addresses and data (prefix = [xx.xx,00,10x,xxx]) (See SRAM and LUT Bits [8:23])</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUT 2 addresses and data (prefix = [xx.xx,00,11x,xxx]) (See SRAM and LUT Bits [8:23])</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Simulation Control Over SPI

System State
- Power up
- Calibration
- Configuration request
- Configuration send
- Timeout setting
- Execution Start
- Execution Stop
- Exception request
- Exception receive
- Digital data output request
- Digital data output receive
- Power down
# Chip Operation Timing

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPI interface clock frequency</td>
<td>16.7 MHz</td>
</tr>
<tr>
<td>Tile controller clock frequency</td>
<td>700 KHz</td>
</tr>
<tr>
<td>Time to complete configuration</td>
<td>2.08 ms</td>
</tr>
<tr>
<td>Time to complete calibration</td>
<td>1.10 ms</td>
</tr>
<tr>
<td>Maximum execution time</td>
<td>1.70 hours</td>
</tr>
</tbody>
</table>
Future Work

- HCDC system test bench
- HCDC compiler
- HCDC software library
- HCDC programming language
HCDC Robotics Workloads
Robots Profiled on Simulators

- Cornell Ranger Simulator
- ROS Gazebo Simulator
  - Turtlebot
  - PR2 mobile base
Candidate Code for HCDC Acceleration

- Sensory input processing
  - Odometry
  - Accelerometer and gyroscope input interpretation
  - Laser rangefinder input interpretation
- Kinematics
  - 3D transforms: translation, rotation
- Mapping
  - Point cloud best fit
- CAVEAT: digital overheads dominate robot runtime
  - ROS kernel
  - Communication routines