Model to Code, Made Simple and Easy

Sebastien Dupertuis
Applications Engineer, Production Code Generation
Applications Engineering Group
MathWorks Switzerland
June 09, 2015
Challenges to bring an idea into real hardware
switch(idea)
{

case 'Applications':
Philips Healthcare MRI scanner

AirSonea device, which connects to a patient's smartphone

Toyota engine

The HB-SIA aircraft on a test flight over San Francisco Bay

Sonova’s hearing aid and cochlear implant solutions

Alstom Grid’s HVDC demonstrator system with power converter modules

http://ch.mathworks.com/company/user_stories/
case 'Programming':
case ‘Hardware’:
case 'Operating Systems':
case 'Standards':
STANDARDS

AUTOSAR

MISRA AC AGC

DO-178B/C

IEC 61508

EN 50128

ISO 26262
default:
  printf("Wrong session?");
}
MBD_Overview();
TEST & VERIFICATION

ANALYSIS – SPECIFICATION – DESIGN

MODEL
- Architecture
- Algorithms
- Schematics
- Environment
- Constraints
- Physical Domains

IMPLEMENTATION
- C, C++
- VHDL, Verilog
- Structured Text
- MCU
- DSP
- FPGA
- ASIC
- PLC
- PAC

INTEGRATION

RESEARCH ACTIVITIES

REQUIREMENTS DOCUMENTS

TEST CASES
Model2Code();
function [symbols, weights] = gainctrl(rxsig, train)
% 1-tap adaptive equalizer using LMS or RLS algorithm

% Equalizer settings
lambda = 0.99;
Delta = 0.1+0j;
weights = 0+0j;

for n = 1:length(rxsig)
    u = rxsig(n);  % received sample
    y = conj(weights) * u;
    if n<=length(train)
        d = train(n);
    else
        d = detect(real(y)) + 1j*detect(imag(y));
    end
% Single-tap RLS
    Delta = 1/(lambda/Delta + u*conj(u));
    G = Delta * u;
    e = d - y;  % symbol estimation error
    weights = weights + G*conj(e);
    symbols(n) = y;
end

MATLAB
function [symbols, weights] = gainctrl(rxsig, train)
% 1-tap adaptive equalizer using LMS or RLS algorithm

% Equalizer settings
lambda = 0.99;
Delta = 0.1+i01;
weights = 0+i01;

for n = 1:length(rxsig)
    u = rxsig(n); % Input
    y = conj(weights); % Initial output
    if n==length(train)
        d = train(n);
    else
        d = detect(real(y)) + 1j*detect(img(y));
    end
    % Single-tap LMS
    Delta = 1/(lambda/Delta + u*conj(u));
    G = Delta * u;
    e = d - y; % Error
    weights = weights + G*conj(e);
    symbols(n) = y;
end

Simulink

Unified representation

MATLAB

Stateflow

Mathematical engines

C Code
C++ Code
HDL Code
PLC Code
Find design errors
Test cases
Fixed-point autoscaling
case ‘Code Generation – Top 5’:
In-the-Loop Verification Methodologies
Software- and Processor- in-the-Loop
SIL and PIL

Non-Real-Time Synchronization with Host at Each Time Step

Execution History
- Logged signal results comparison
- Code coverage
- Execution timing
Hardware-in-the-Loop
HIL, Rapid Prototyping

Code Generation
Hard Real-Time Execution
Logging and Tuning via Host
FPGA-in-the-Loop
FIL, Test Bench Simulation
Incremental Build Process
Incremental Build Process

- Significantly saves time
- Only build blocks that have changed
- Helps with partitioning and componentization
- Scalability!
Simulink Data Dictionary
Manage data outside of base workspace

- Componentization
- Scalability and performance
- Change tracking and differencing
- Integration with Simulink Projects
- Code generation
Profiling of Generated Code
Measure Execution Time

Identify hot spots, worst-case execution

- Supports
  - SIL and PIL
  - Tasks and functions
  - HTML reports
Intellectual Property Protection
Password Protected Models

Protect design IP for models and generated code

- **Support options**
  - Simulation: Allow Accelerator mode
  - Code generation: Include obfuscated code to support code generation
  - Read-only view: Web view of model
  - Password protection: Access protected by password
case ‘Targets’:
Hardware Support Packages

- HW Support Packages are:
  - Downloadable from MathWorks websites
  - Available for free with required base product
  - Supported by technical support

- HW Support Package manages:
  - Licenses
  - 3rd-party software installation
  - Hardware setup
Services

- Automate compile, build, and download
- Integrate device drivers and RTOS with Simulink
- Optimize code replacements to your target
- Verify and validate code execution results
case 'Getting FREEd':
}
Programming an heterogeneous system

Zynq Platform