rtb_either = power_window_con_B.passenger_control_b || power_window_con_B.passenger_control_a;

/* Logic: '<S13>/allow_action' incorporates:
* Inport: '<Root>/driver_up'
* Logic: '<S13>/overrule'
*/

rtb_temp34 = power_window_con_U.driver_up && !(rtb_either);
Introduction

- Increasing complexity of embedded systems
- Complexity
  - Intricacy
  - Size
- Raising the Level of Abstraction
- Compilers to handle the complexity because of size
Agenda

- The System Design Challenge
- Software Design Flow
- Hardware Design Flow
- Summary
The System Design Challenge

- We design, simulate, and validate system models and algorithms in MATLAB® and/or Simulink®

- How can we implement and verify designs on DSPs and GPPs?

- How can we implement and verify designs on FPGAs and ASICs?
Integrated Design Flow for Embedded Software and Hardware

- Design, simulate, and validate system models and algorithms in MATLAB and Simulink
- Automatically generate C and HDL
- Verify hardware and software implementations against the system and algorithm models
Agenda

- The System Design Challenge
- **Software Design Flow**
- Hardware Design Flow
- Summary
Integrated Design Flow for Embedded Software

- Implementation with automatic C code generation

- Implementation
  - Floating- and fixed-point code
  - Integration with downstream IDEs and tools
  - Links to verification
  - Device drivers
  - Optimization options
Representative Application – Video

- Video system design and implementation
  - Encapsulates challenges: complexity, convergence, time-to-market
  - Sophisticated algorithms
  - Floating- and fixed-point issues
  - DSP or FPGA/ASIC implementations

- Design flows and steps shown directly applicable to other signal processing applications
Example: Video Edge Detection

- Floating point video edge detection system based on Prewitt algorithm
- Compositing original image with detected edges
- Utilizes blocks from Video and Image Processing Blockset
Converting to Fixed-Point

- Polymorphic blocks capable of floating- and fixed-point operation
- 8-bit input datatype – blocks inherit fixed-point data
- Simulink Accelerator provides fast fixed-point simulation

8-bit fixed-point input
Automatic Code Generation for Implementation on GPPs and DSPs

- Fixed-point model
- Code generation options and preferences
- Select target or flavor of generated code
Code Generation Report

Code Generation Report for vipedge_fixpt_win

Readable, commented code

Links to code files from HTML report

Links to Simulink model from generated code

HTML report
Video Edge Detection
Embedded Software System on TI 6000™
Target Code Generation Options

- **High-Speed RTDX**
- **Incorporate DSP/BIOS**
- **Inline Signal Processing Blockset functions**
Code Execution on Target and Profiling

- Build and execute
  - Auto-generate C and ASM
  - Integrate RTOS and scheduler
  - Create full CCS project
  - Invoke compiler, linker, and download code
  - Run on target

- Profile code performance

System profiling includes entire DSP application code

Subsystem profiling
Code Optimization Options

- Utilize target-specific blocks
  - C-callable assembler libraries
  - Simulate bit-true in Simulink
  - Generate calls to hand-optimized assembler libraries
  - Highly optimized implementation of core functionality
  - C62x and C64x fixed-point DSPs
- Manual optimization by user
Design Verification and Visualization:
MATLAB as software test bench

```matlab
% Initialize motion threshold
global newThresh
lastThresh = 0;
newThresh = 1.5e5;

% Connect to CCS
CCS_Obj = connectToCCS(modelName);
saved_visibility = CCS_Obj.isvisible;
CCS_Obj.visible(1);

% Load application
loadApp(modelName, CCS_Obj);

% Run application
fprintf('Running application: %s\n', modelName);
CCS_Obj.run;

% Connect to the target
userPrompt = sprintf('Please enter the IP address or the host name of the %s board: ',
hostName = input(userPrompt, 's');
port = 49000;
fprintf('Connecting to TCP/IP server at: "%s:%d"\n', hostName, port)
connf = tcpip(hostName, port);
set(connf, 'OutputBuffersize', 64000);
set(connf, 'InputBuffersize', 64000);
try
    fopen(connf);
except
    fprintf('
Cannot establish TCP/IP connection to "%s:%d"\n', hostName, port);
    fprintf('Terminating demo.\n');
    return;
end
set(connf, 'ByteOrder', 'littleEndian');
fprintf('Established TCP/IP connection to "%s:%d"\n', hostName, port);
```

Visualize and debug embedded software with MATLAB

Input video

Captured video

MATLAB script
(test bench)
Design Verification and Visualization:
Simulink as software test bench

Processor and hardware-in-the-loop testing, simulation, visualization, and verification of embedded software with Simulink
Review: Integrated Design Flow for Embedded Software

- Drive system development with an executable specification
- Quickly create complete working code base
- Use code profiles to identify and optimize bottlenecks
- Verify code with Links to IDEs and processors
Agenda

- The System Design Challenge
- Software Design Flow
- Hardware Design Flow
- Summary
Integrated Design Flow for Hardware (FPGA and ASIC)

- Design elaboration

- Implementation
  - HDL code generation (VHDL and Verilog)
  - test bench generation
  - Links to verification
  - Integration with synthesis tools
What We Will Show In This Case Study

- Behavioral modeling and simulation
- Fixed-point modeling and simulation
- Design elaboration
- HDL generation
- Co-simulation using Link for ModelSim

Hardware Case Study:
Video Edge Detection on an FPGA

Live MATLAB Demo
Executable Specification
Design Elaboration

Automatically generate Verilog or VHDL code from elaborated models
HDL Code Generation Using GUI

- Select subsystem, target language, directory
- Select output options
- Check model for errors
- Generate HDL Code
More Code Generation Options

Select reset and clock options

Set language-specific options: input/output datatypes, timescale directives, …
Automatically Generate Test Bench

Self-checking HDL test bench compares Simulink results to HDL results
Co-simulate Generated HDL

HDL code executing on ModelSim simulator
Review:
Integrated Design Flow for Hardware

- Drive system development with an executable specification

- Quickly create complete working HDL code base and test benches

- Verify code with Links to RTL simulators and synthesis tools
Agenda

- The System Design Challenge
- Software Design Flow
- Hardware Design Flow
- Summary
Summary

- Accelerate development using Model-Based Design
  - Generate
    - Real-Time Workshop
    - Simulink HDL Coder
  - Verify
    - Link for Cadence Incisive

- Design and verify software and hardware from MATLAB and Simulink