Model-Based Design of Embedded Signal Processing Systems Using Simulink®

Altera SOPC World 2004

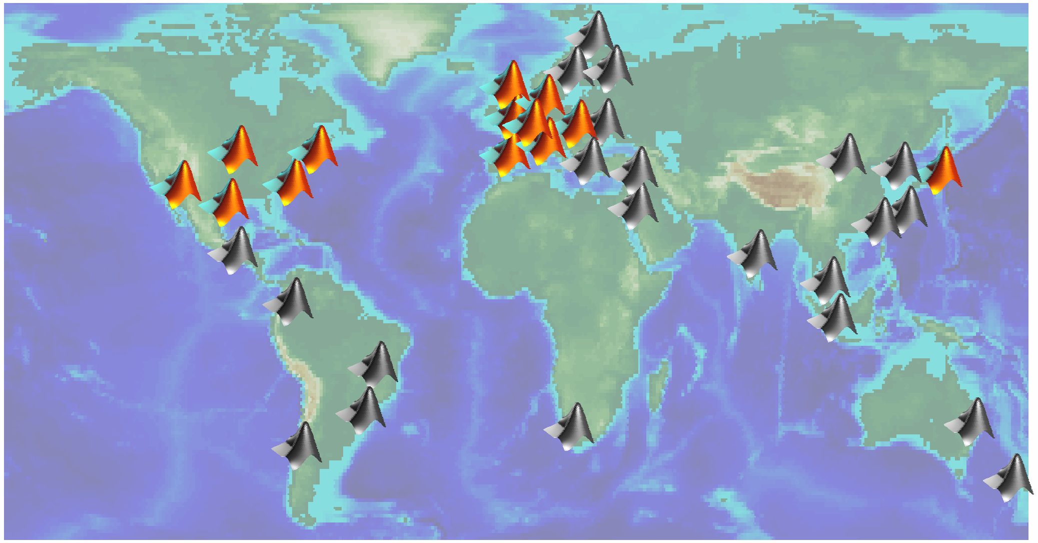
<Name of presenter here>

Agenda

- Model-Based Design of Embedded Systems
  - Challenges in DSP system design
- Simulink and Blocksets
  - Quick Simulink demo
  - Video surveillance demo
- Hardware Implementations
  - Implementation on DSPs and Altera FPGAs

The MathWorks at a Glance

- **Headquarters:**
  Natick, Massachusetts USA
- **USA:**
  California, Michigan,
  Washington DC, Texas
- **Europe:**
  UK, France, Germany,
  Switzerland, Italy,
  Spain, Benelux
- **Asia-Pacific:**
  Korea
- **Worldwide training and consulting**
- **Distributors in 25 countries**



Earth's topography on an equidistant cylindrical projection, created with the MATLAB Mapping Toolbox

Key Industries

Core

- Aerospace and Defense
- Automotive
- Communications, Electronics, Semiconductor, Computers and Office Equipment
- Education

Emerging

- Biotech, Pharmaceutical and Medical
- Financial Services
- Industrial Equipment and Machinery
- Instrumentation

Ongoing

- Chemical and Petroleum
- Earth and Ocean Sciences
- Utilities and Energy

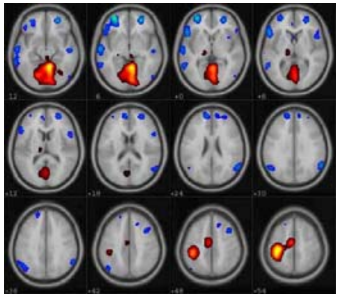
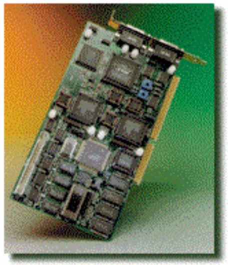
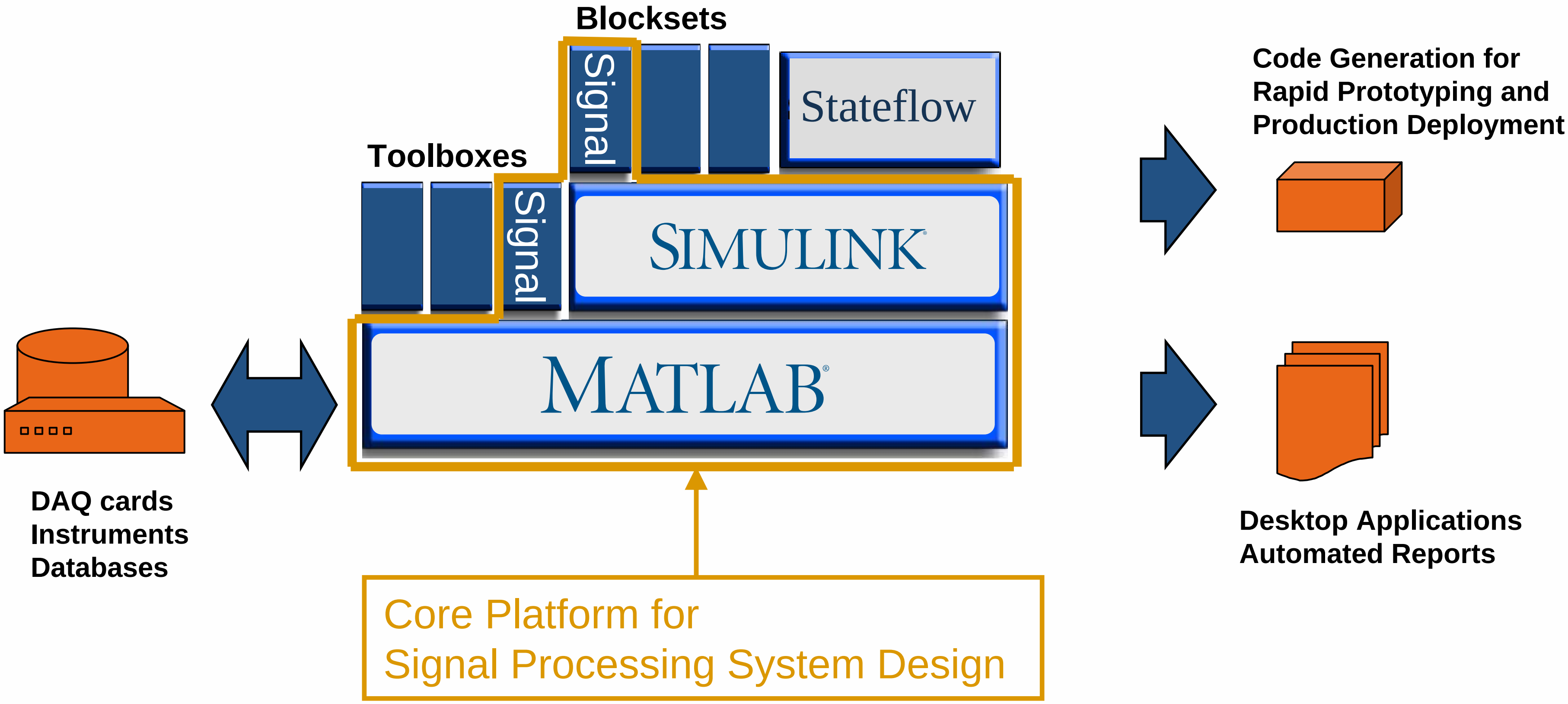


Figure No. 2
File Edit View Insert Tools Window Help
International Business Machines
Price ($)
115
110
105
100
95
90
12/31
01/10
01/20
01/30
02/09
02/19



The MathWorks Product Family



Blocksets

Signal

Stateflow

Toolboxes

Signal

SIMULINK

MATLAB

DAQ cards
Instruments
Databases

Core Platform for
Signal Processing System Design

Code Generation for
Rapid Prototyping and
Production Deployment

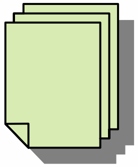
Desktop Applications
Automated Reports

Challenges in Embedded System Design

- Handle Increasing complexity
- Design team integration
- Reduce Time-to-Market

Problems with Traditional Development

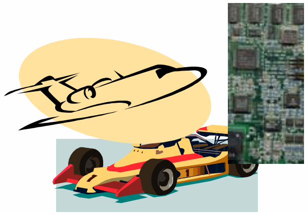
Requirements and Specifications



Text-based

- Prevents rapid iteration

Design



Physical prototypes

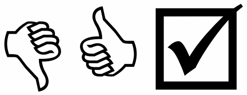
- Incomplete and expensive

Implementation

Manual coding

- introduces human error

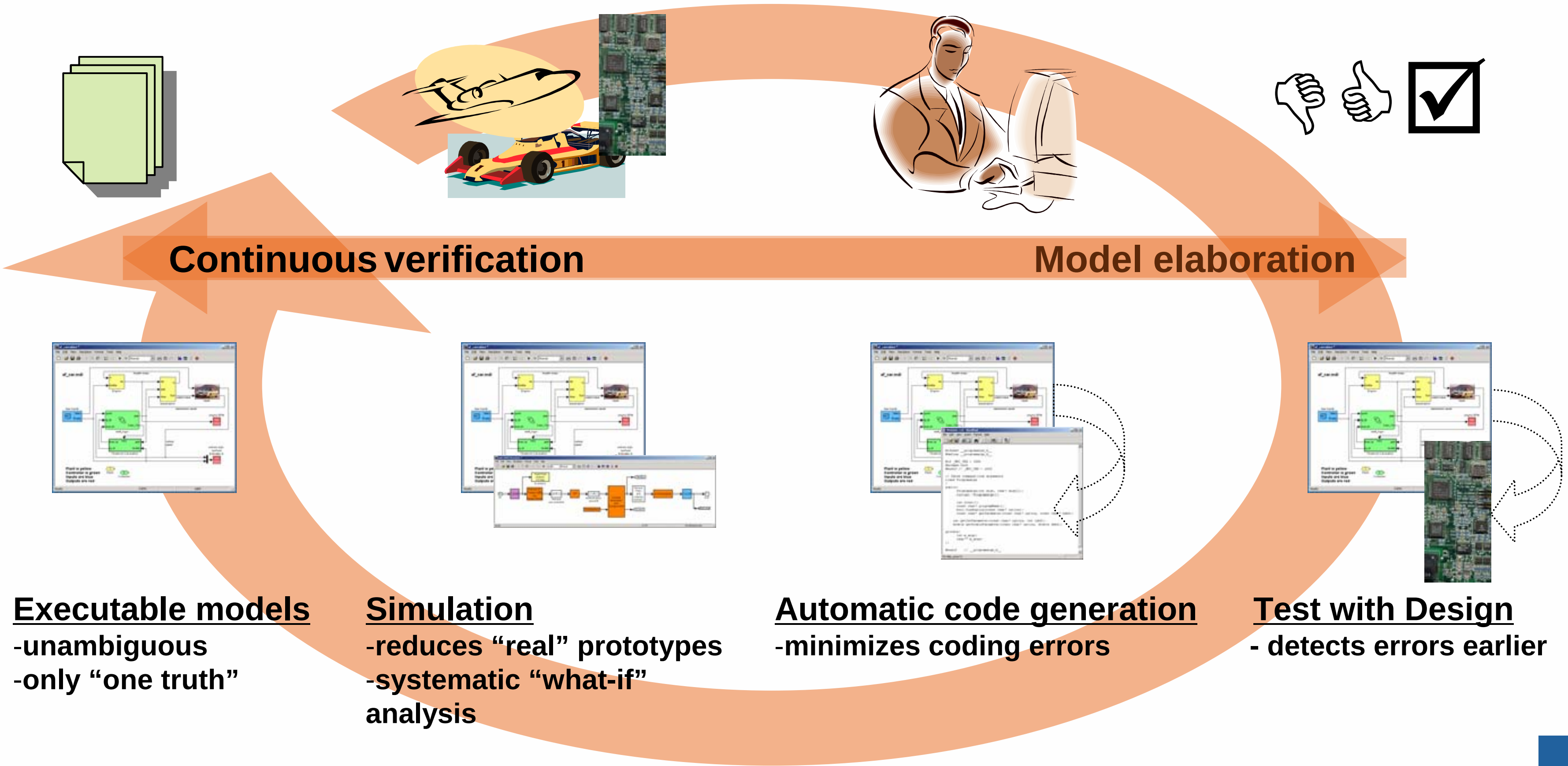
Test and Verification



Traditional testing

- errors found too late in the process

Advantages of Model-Based Design



Requirements and Specifications

Design

Implementation

Test and Verification

Continuous verification

Model elaboration

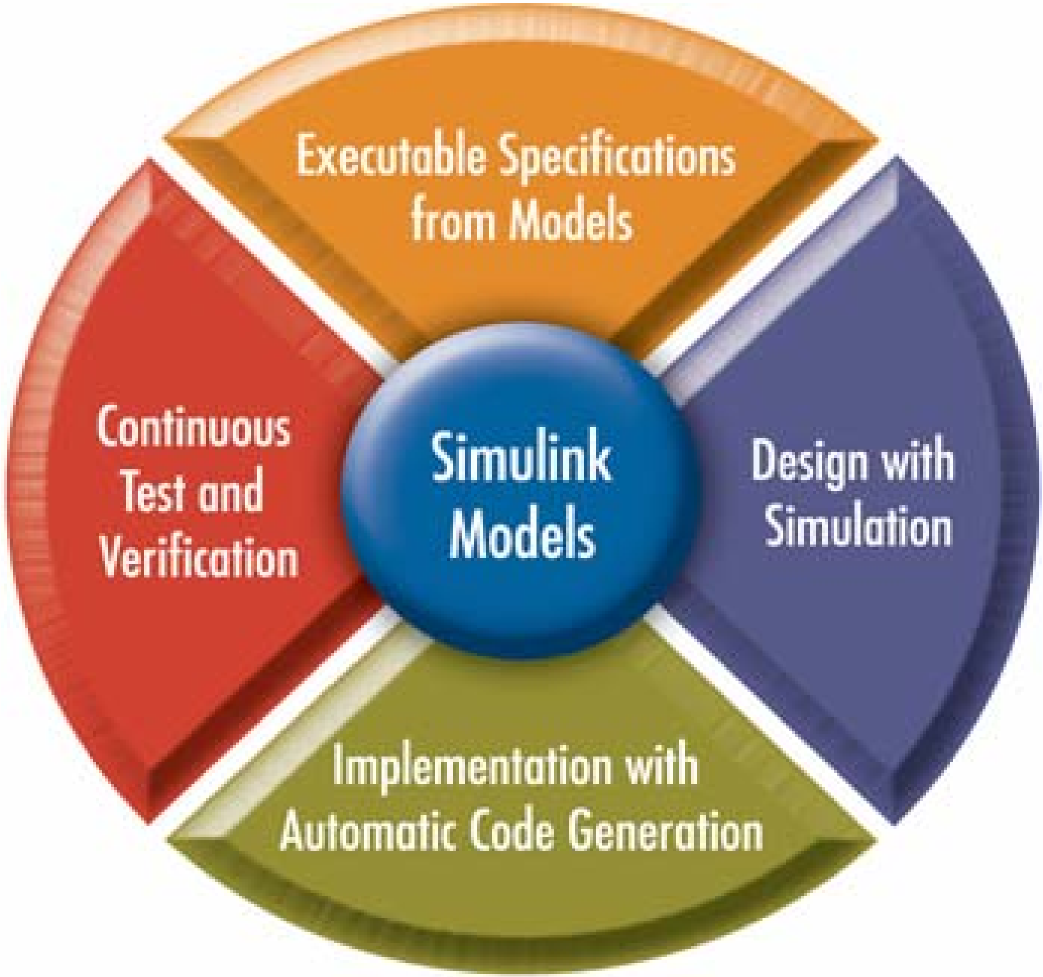
Executable models
-unambiguous
-only “one truth”

Simulation
-reduces “real” prototypes
-systematic “what-if” analysis

Automatic code generation
-minimizes coding errors

Test with Design
- detects errors earlier

Model-Based Design with Simulink


Executable Specifications from Models
Continuous Test and Verification
Simulink Models
Design with Simulation
Implementation with Automatic Code Generation

Model-Based Design Allows You to Overcome Design Challenges:

- **Handle Design Complexity**
  - Reuse IP, Simulate at high speeds, collaborate with multiple design teams, utilize system-level design abstraction
- **Design team integration**
  - Analog/Mixed-Signal, digital hardware, DSP S/W, control S/W designed in one model
  - Co-design and partition HW and SW components
- **Reduce Time-to-Market**
  - Generate code automatically for HW and SW
  - Accelerate verification using executable specification

User stories: RealTek

RealTek Gains 50% of Market Share with a New Audio Chip Designed with MathWorks Tools

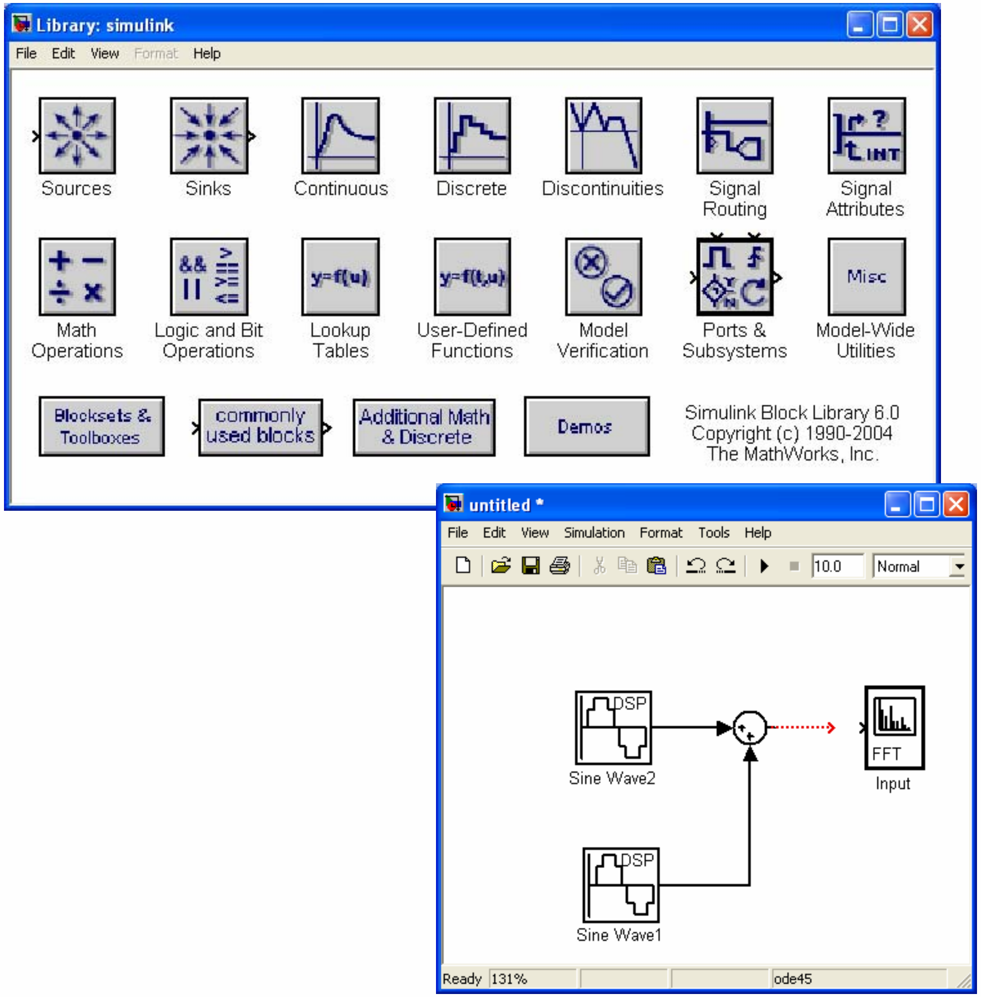
Challenge To unify different engineering disciplines on a single development platform

Solution Standardize on MathWorks tools to streamline the design process and enable analog and DSP designers to work together

Results
- 50% market share in first year of product release.
- High return on investment.
- Improved collaboration and reduced design time.

Products Used
- MATLAB
- Simulink
- DSP Blockset
- Fixed-Point Blockset
- Optimization Toolbox
- Signal Processing Toolbox

Simulink Tutorial: Model Construction



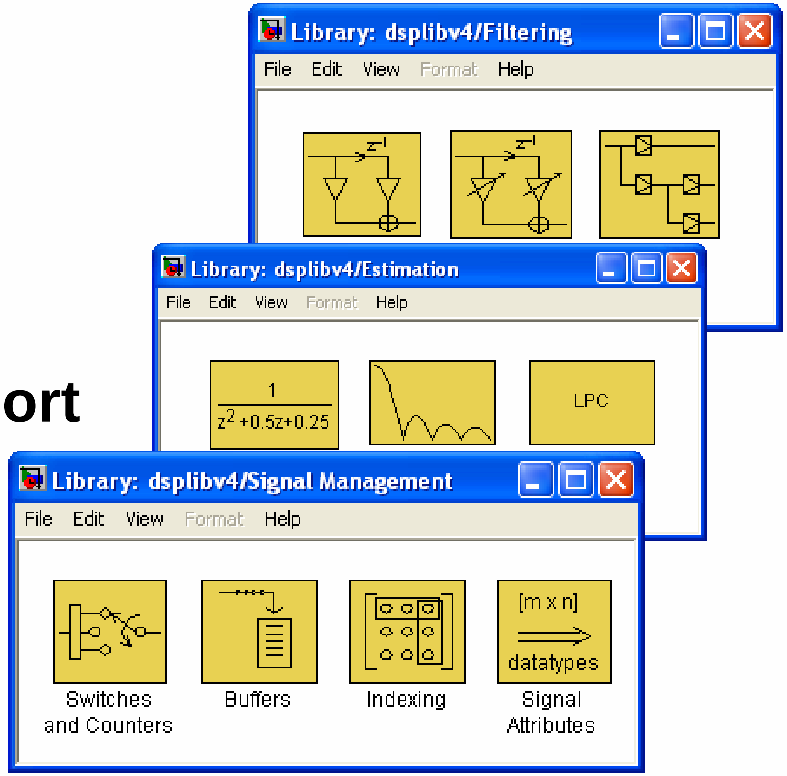
Simulink Block Library 6.0
Copyright (c) 1990-2004
The MathWorks, Inc.

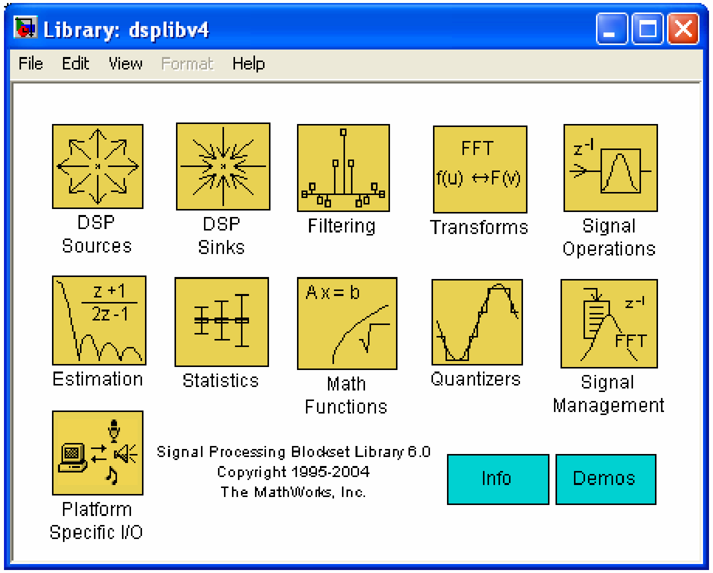
- Drag and drop
- Connect

- Digital
  - Fast frame-based simulation
- Analog
  - Variable-step numerical integration solvers
  - Zero-crossing detection

Signal Processing Blockset

- **Streaming data**
- **Multi-rate systems**
- **Transforms, filters, estimators**
- **Enables frames in Simulink**
- **Fixed- and Floating-Point Support**


Library: dsplibv4/Filtering
File Edit View Format Help
Library: dsplibv4/Estimation
File Edit View Format Help
1
z2 +0.5z+0.25
LPC
Library: dsplibv4/Signal Management
File Edit View Format Help
[m x n]
datatypes
Switches and Counters
Buffers
Indexing
Signal Attributes



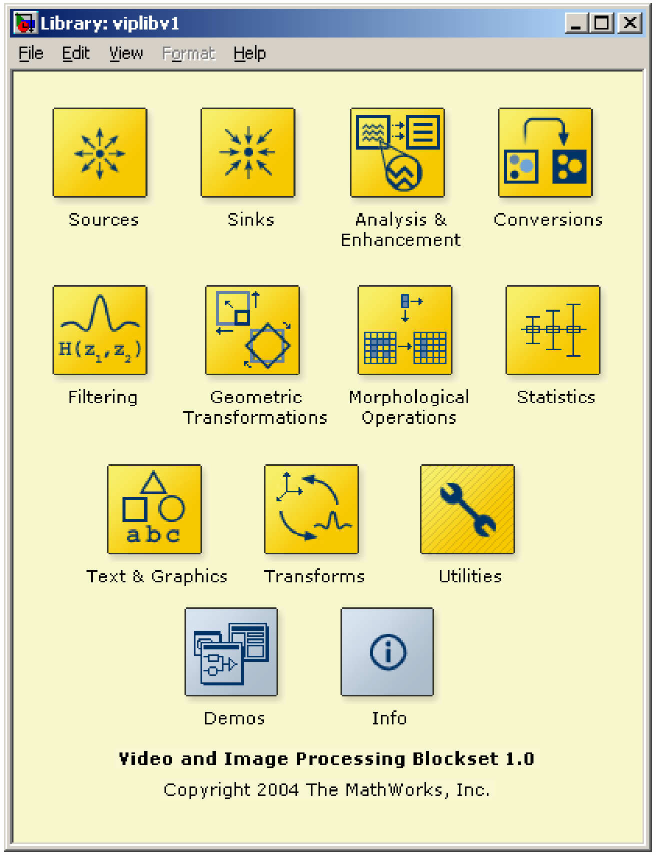
Signal Processing Blockset Library 6.0
Copyright 1995-2004
The MathWorks, Inc.

Example 1 - Envelope Detection

>> dspwwvlib.mdl

>>dspwwv

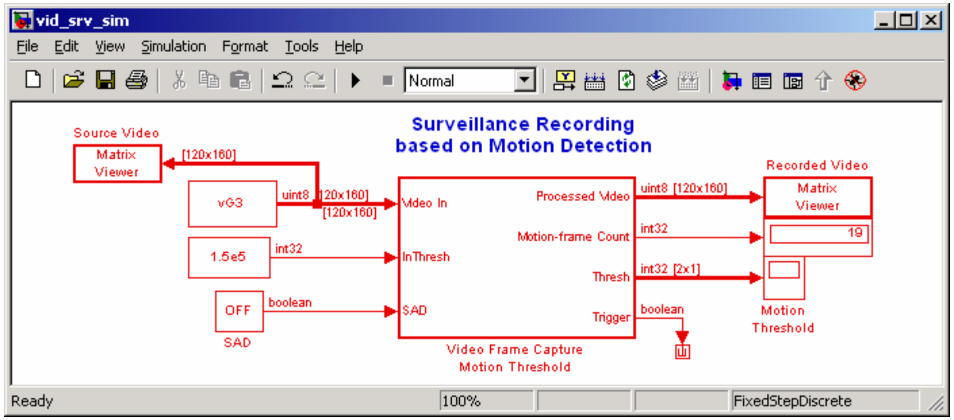
Example 2 - Video Surveillance Systems using the Video and Image Processing Blockset



Video and Image Processing Blockset 1.0

Copyright 2004 The MathWorks, Inc.

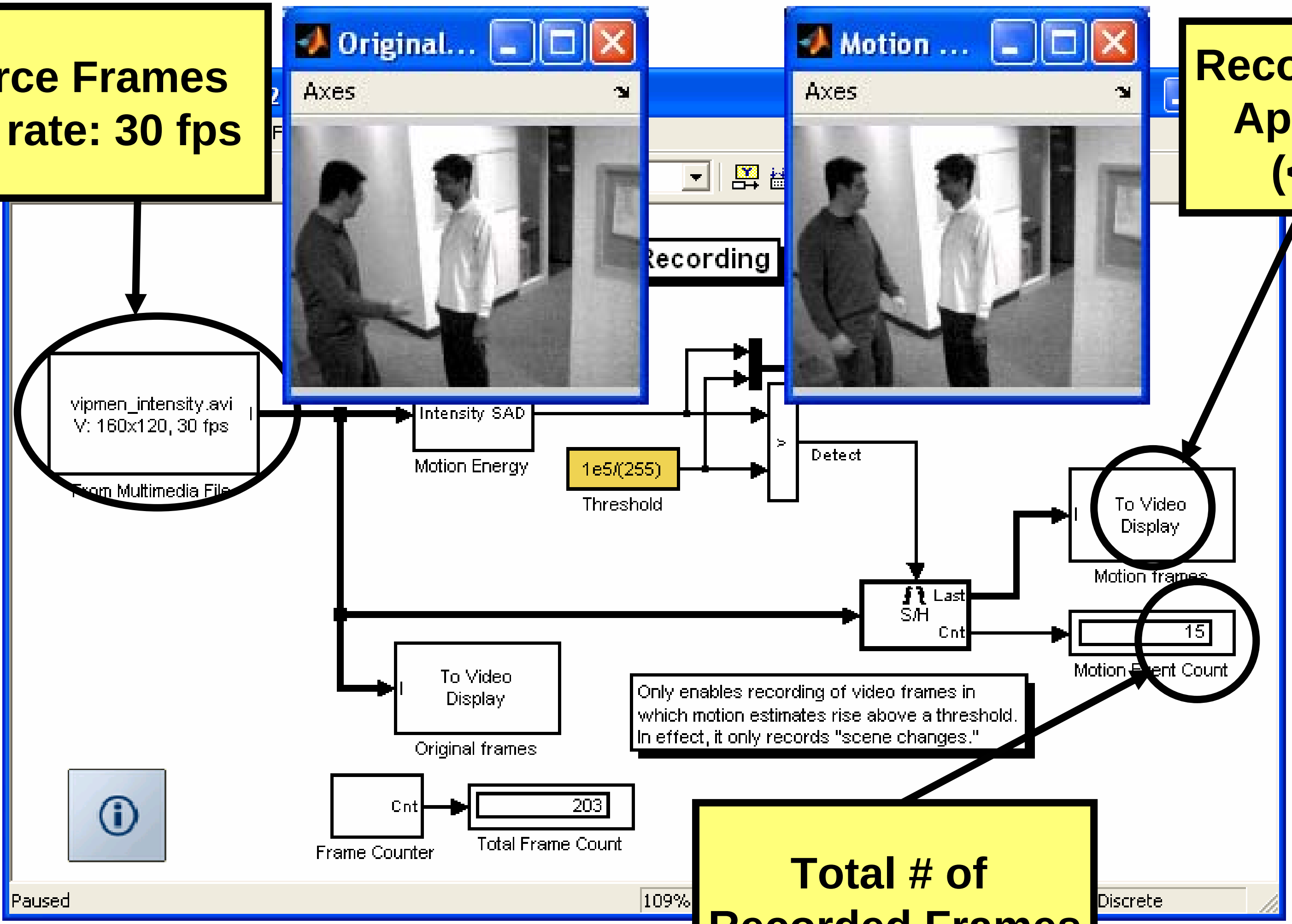
Fixed-point video surveillance system based on Sum of Absolute Differences motion detection



Example 2: Video Surveillance System

Source Frames
High rate: 30 fps

Recorded Frames
Aperiodic rate
(<< 30 fps)

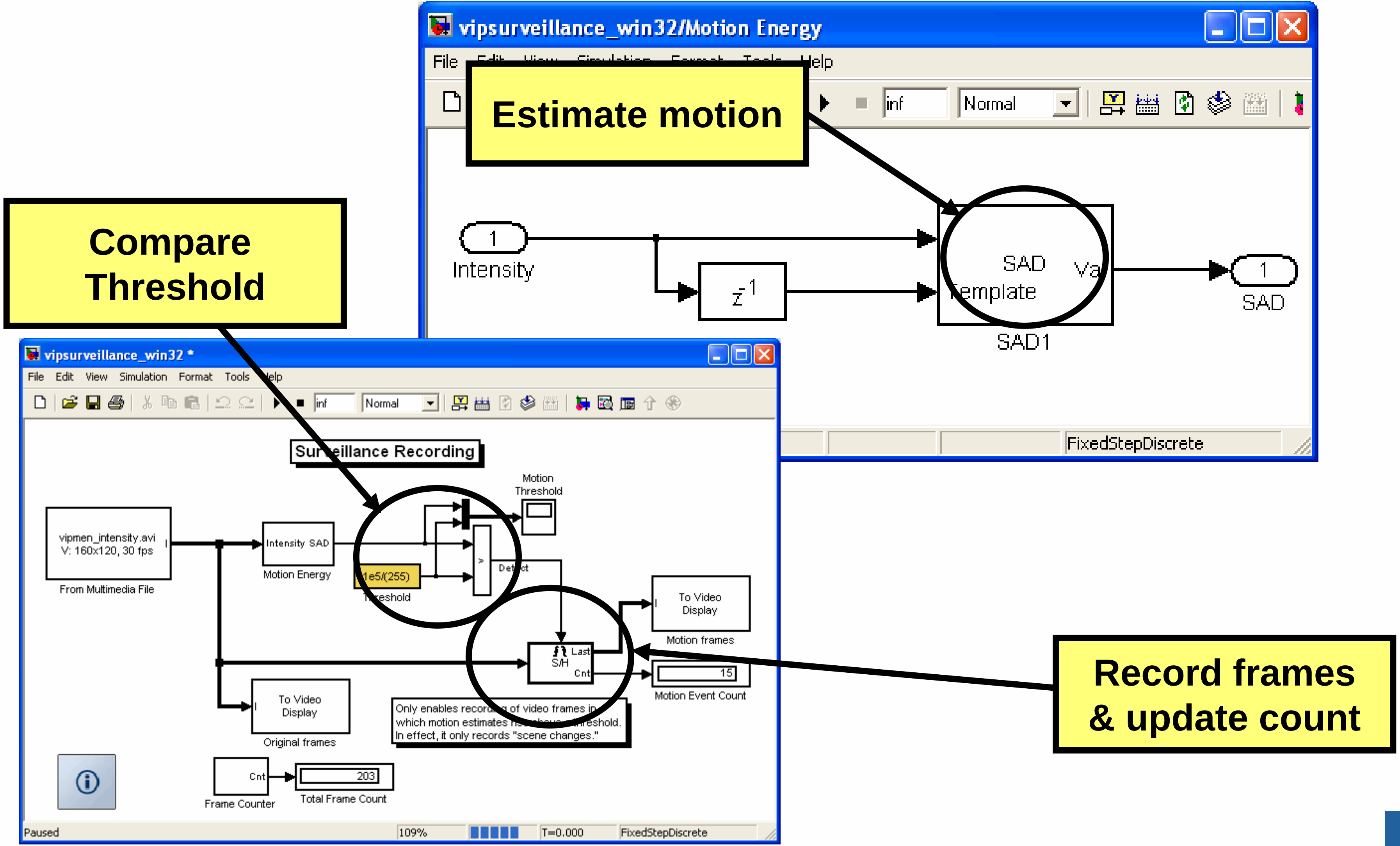


Only enables recording of video frames in which motion estimates rise above a threshold. In effect, it only records "scene changes."

Total # of
Recorded Frames

```
>> vipsurveillance_win32
```

Motion Estimation: Sum of Absolute Differences



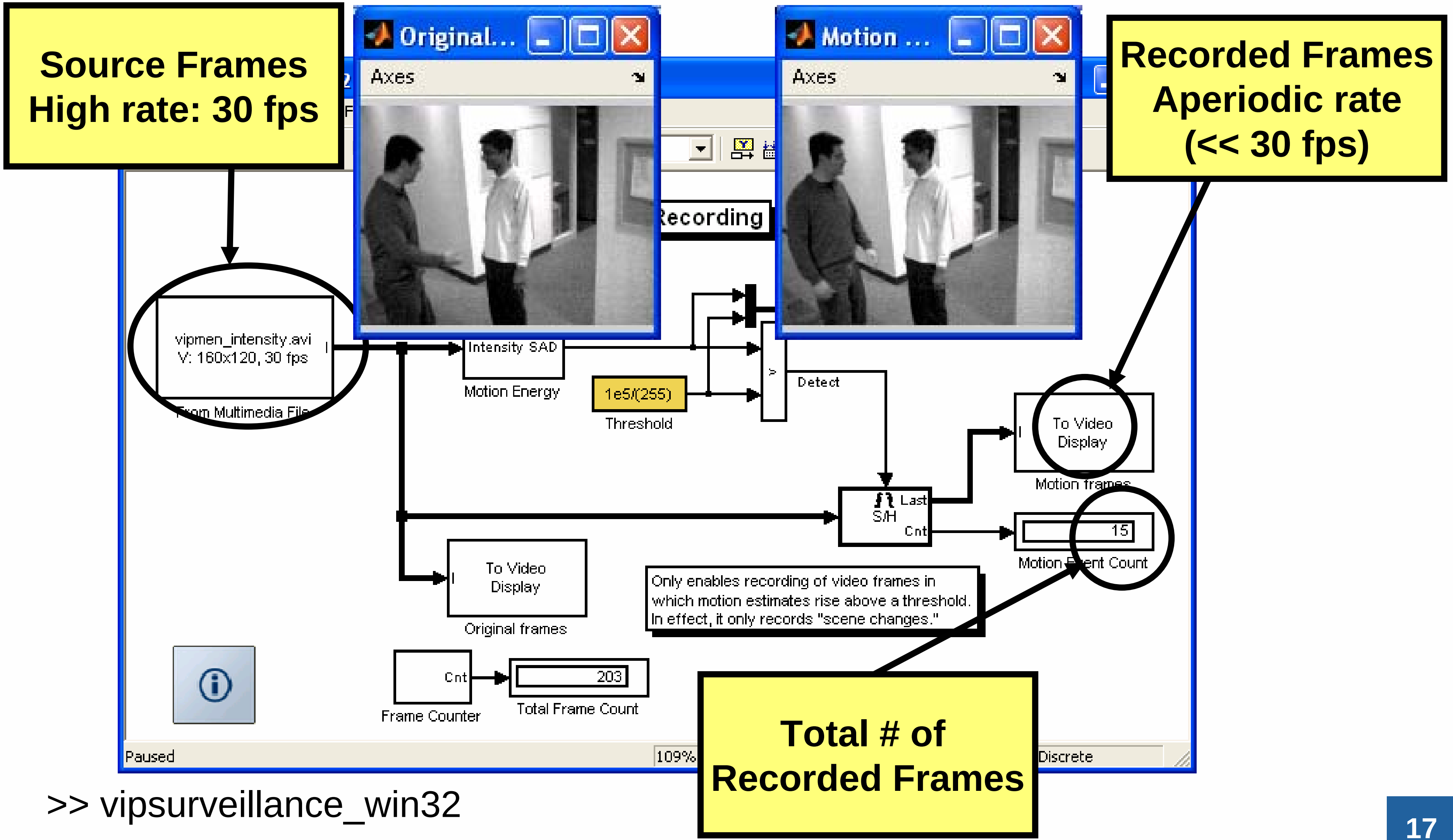
Estimate motion

Compare Threshold

Record frames & update count

Only enables recording of video frames in which motion estimates rise above a threshold. In effect, it only records "scene changes."

Example 2: Video Surveillance System



Source Frames
High rate: 30 fps

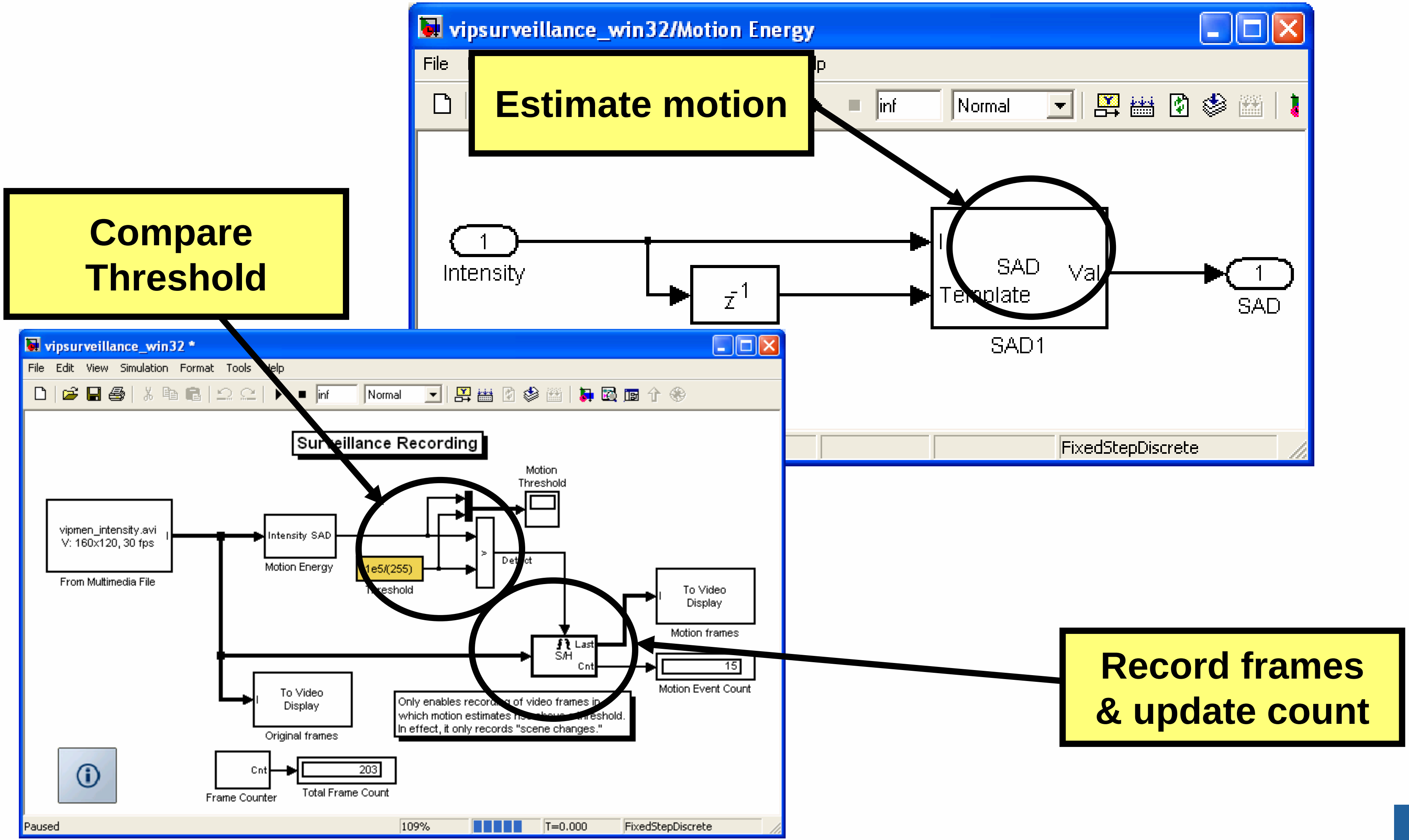
Recorded Frames
Aperiodic rate
(<< 30 fps)

Total # of
Recorded Frames

Only enables recording of video frames in which motion estimates rise above a threshold. In effect, it only records "scene changes."

>> vipsurveillance_win32

Motion Estimation:
Sum of Absolute Differences



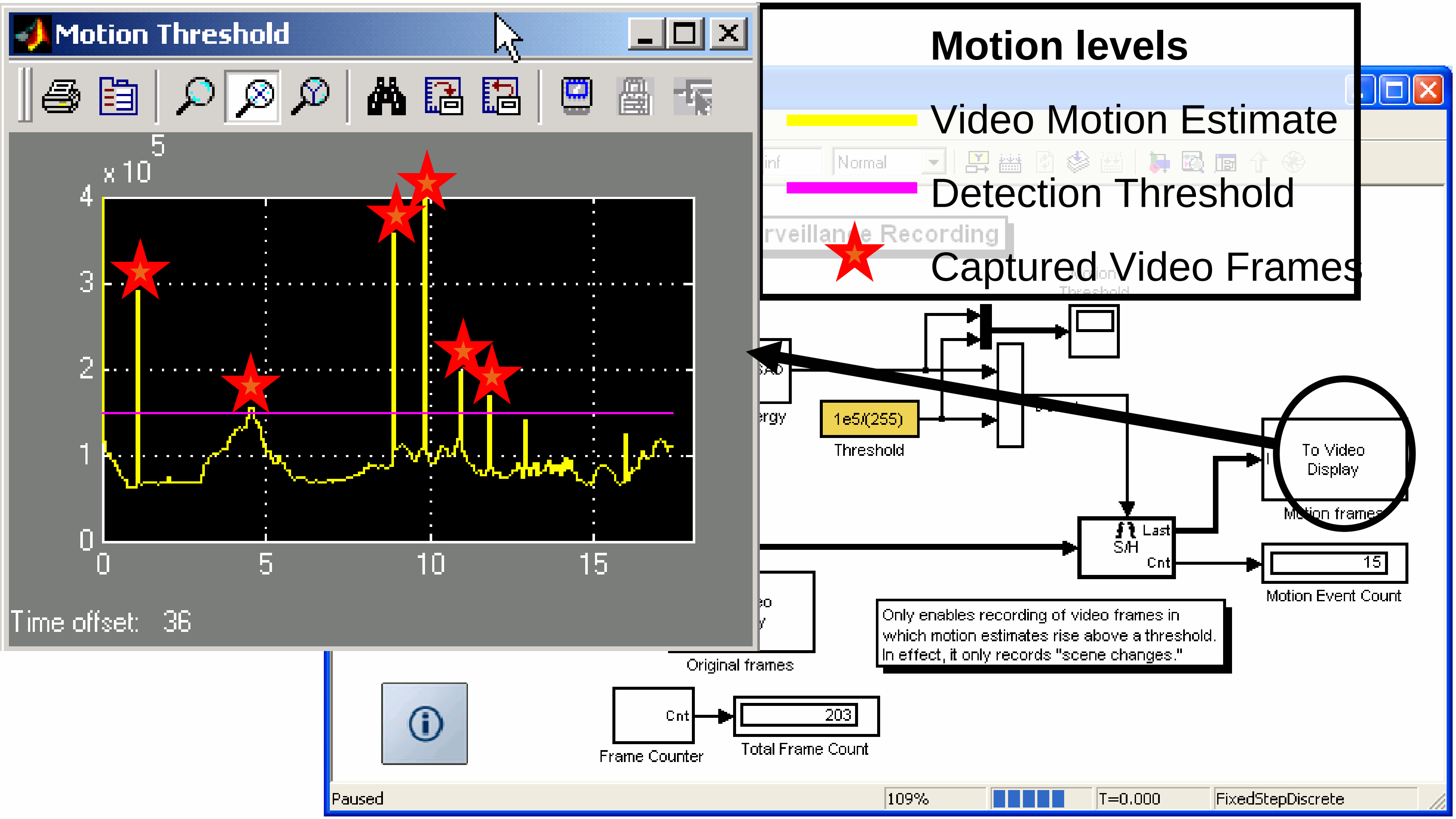
Estimate motion

Compare Threshold

Record frames & update count

Only enables recording of video frames in which motion estimates is above threshold. In effect, it only records "scene changes."

Motion Detection with Thresholding



Only enables recording of video frames in which motion estimates rise above a threshold. In effect, it only records "scene changes."

Embedding Signal Processing Applications on DSPs and FPGAs

Production Code Generation

- Based on Real-Time Workshop code generation engine
- Generated code is ANSI C – efficient, readable, editable
- Supports and utilizes <u>Real-Time Workshop Embedded Coder</u>

- Proven automatic code generation technology for critical applications

TOYOTA

Visteon

DENSO

JAGUAR

BAE SYSTEMS

Honeywell

LOCKHEED MARTIN
We never forget who we're working for™

Production Code Generation

Successful in Automotive and Aerospace industries

"Visteon Powertrain has demonstrated that model-based software development can generate quality software in less time, and the automatic code ROM & RAM sizes are equal to or better than hand written code.*"

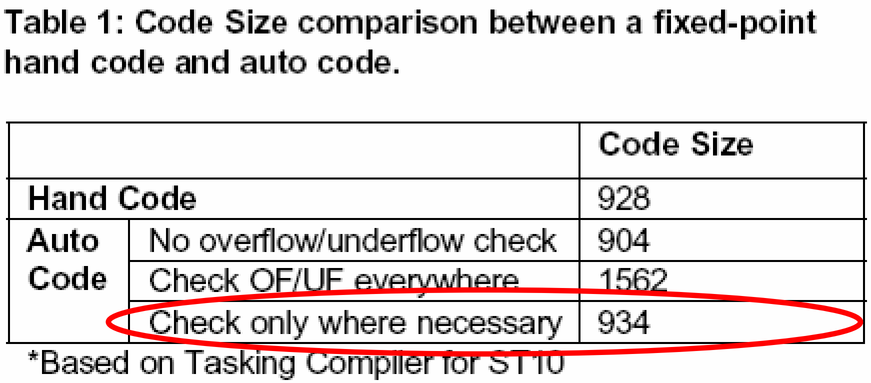


Table 1: Code Size comparison between a fixed-point hand code and auto code.

		Code Size
Hand Code		928
Auto Code	No overflow/underflow check	904
	Check OF/UF everywhere	1562
	Check only where necessary	934

*Based on Tasking Compiler for ST10

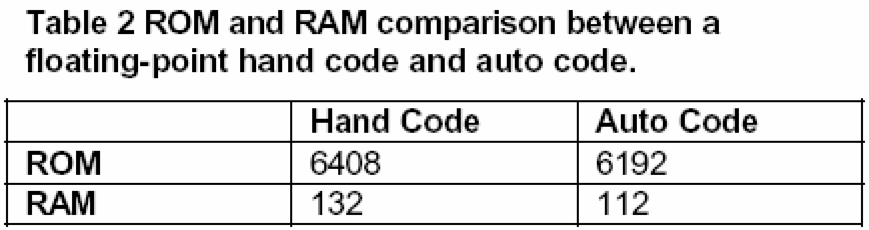


Table 2 ROM and RAM comparison between a floating-point hand code and auto code.

	Hand Code	Auto Code
ROM	6408	6192
RAM	132	112

* Multi-Target Modeling for Embedded Software Development for Automotive Applications
Grantley Hodge, Jian Ye and Walt Stuart, Visteon Corporation
2004 SAE World Congress, Detroit, MI. March 8-11, 2004

Embedded Target for TI C6000 DSP is…

… 1. a tool for production code generation

- Processor-specific, optimized
- Simulink blocks and optimized libraries (FIR, FFT, …)

…2. a means for project automation

- Processor-specific, automatic
- APIs for CCS IDE, Compiler/Linker

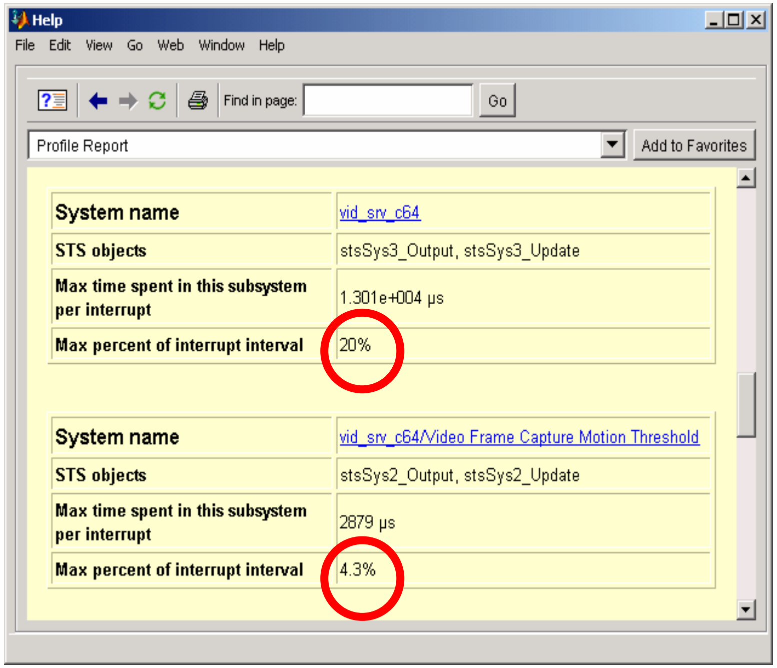
… 3. a platform for rapid prototyping

- Target-specific, integrated
- Simulink hardware blocks and device drivers (ADC, DAC, RTDX, daughter cards)

1. Production Code Generation

a) How good is the generated code for TI C6000?

- Code generation philosophy for C6000 DSPs:
  - Generates efficient, portable, readable, editable code
  - Supports code profiler to help <u>identify code performance bottlenecks</u>
    - code segments that provide highest return on optimization



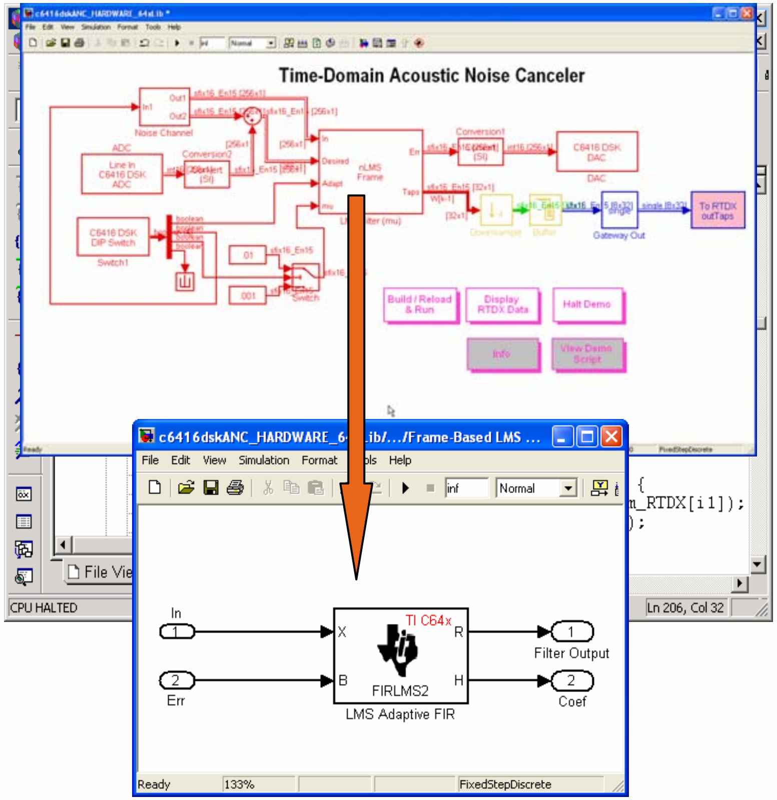
System name	vid_srv_c64
STS objects	stsSys3_Output, stsSys3_Update
Max time spent in this subsystem per interrupt	1.301e+004 µs
Max percent of interrupt interval	20%

System name	vid_srv_c64/Video Frame Capture Motion Threshold
STS objects	stsSys2_Output, stsSys2_Update
Max time spent in this subsystem per interrupt	2879 µs
Max percent of interrupt interval	4.3%

1. Production Code Generation

b) How can I further optimize the code?

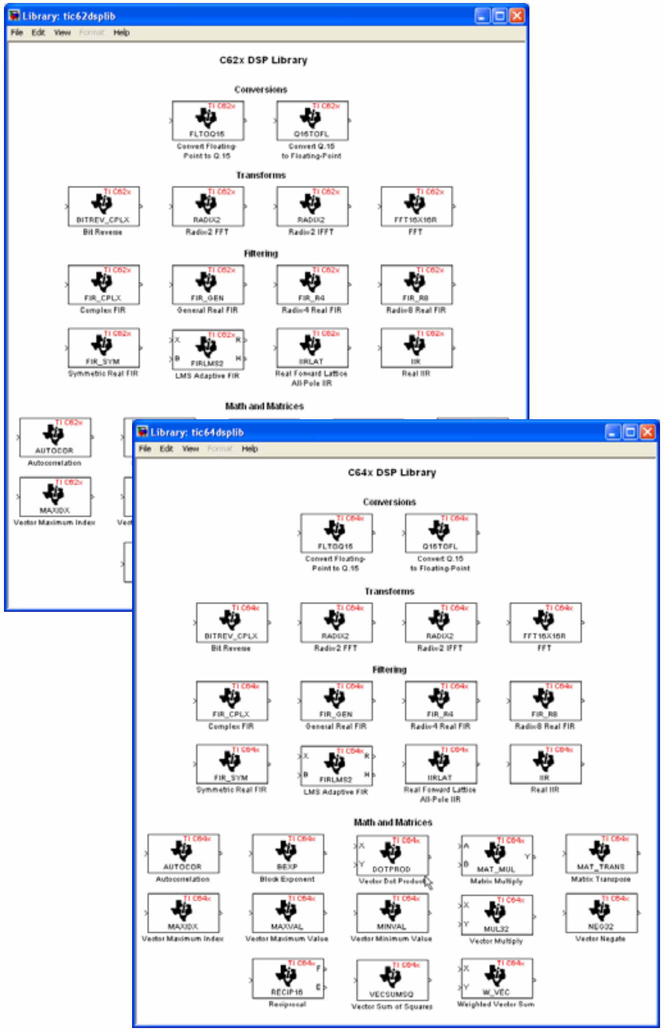
- Provides alternate methods for code optimization
  - Manual optimization by user
    Click on link in profile report to jump to relevant code section
  - Target-specific blocks
  - Engineering services



1. Production Code Generation

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



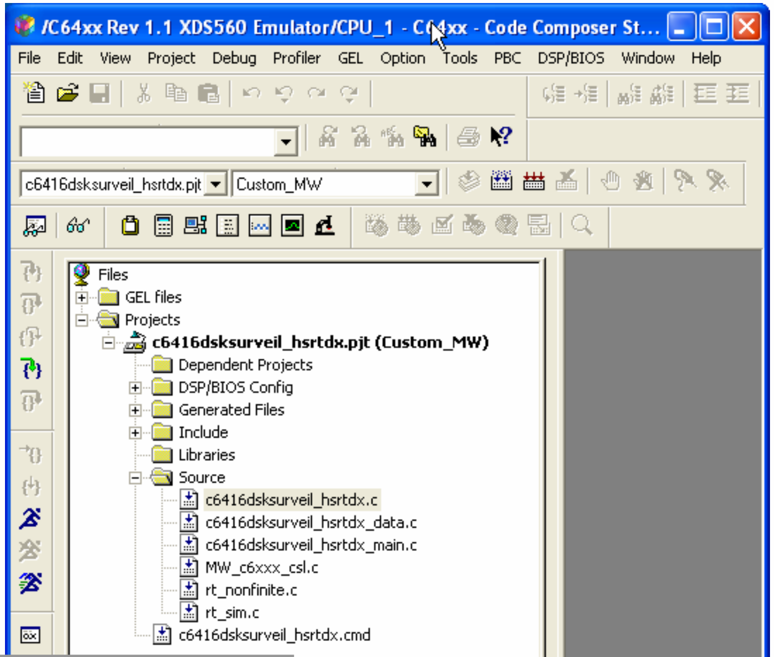
1. Production Code Generation

- Recap of code generation with Embedded Target for TI C6000
  - Quickly create a complete, working code base
  - Utilize code profiler to help identify any performance bottlenecks in generated code
  - Choose from several approaches for code optimization

…optimize only when and where necessary…

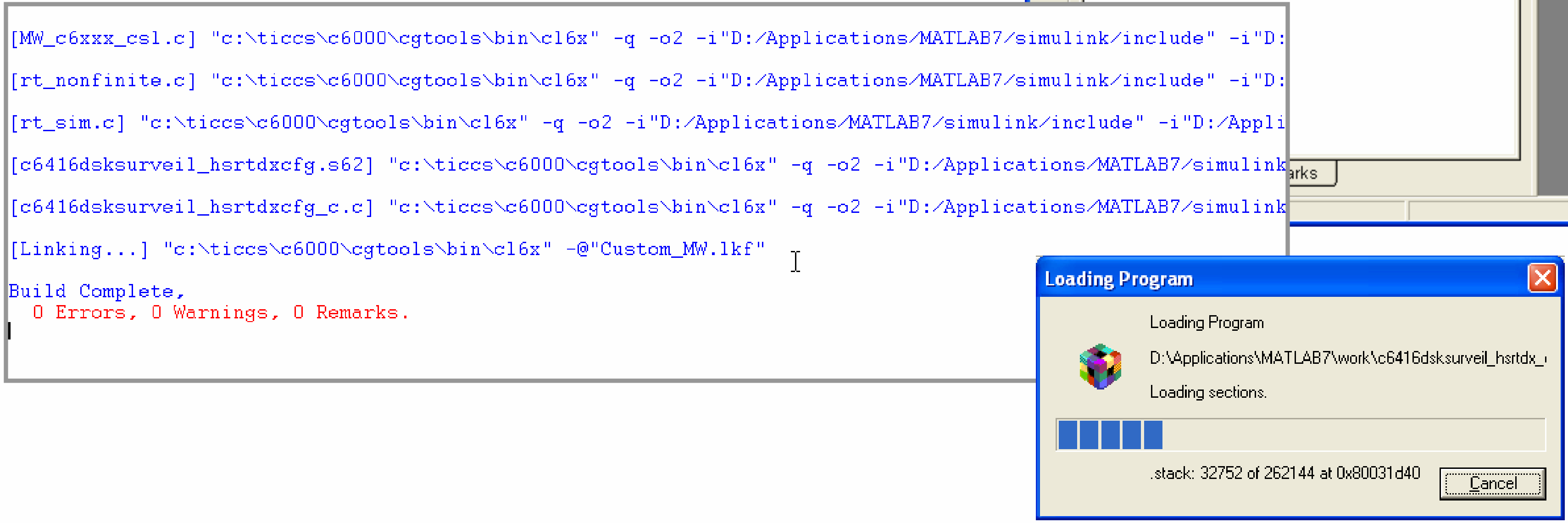
2. Project Automation

- Create and populate CCS project
- Automate compile/link/download



/C64xx Rev 1.1 XDS560 Emulator/CPU_1 - C64xx - Code Composer St...
File Edit View Project Debug Profiler GEL Option Tools PBC DSP/BIOS Window Help
c6416dsksurveil_hsrtdx.pjt
Custom_MW

```
Files
  GEL files
  Projects
    c6416dsksurveil_hsrtdx.pjt (Custom_MW)
      Dependent Projects
      DSP/BIOS Config
      Generated Files
      Include
      Libraries
      Source
        c6416dsksurveil_hsrtdx.c
        c6416dsksurveil_hsrtdx_data.c
        c6416dsksurveil_hsrtdx_main.c
        MW_c6xxx_csl.c
        rt_nonfinite.c
        rt_sim.c
      c6416dsksurveil_hsrtdx.cmd
```



```
[MW_c6xxx_csl.c] "c:\ticcs\c6000\cgtools\bin\cl6x" -q -o2 -i"D:/Applications/MATLAB7/simulink/include" -i"D:
[rt_nonfinite.c] "c:\ticcs\c6000\cgtools\bin\cl6x" -q -o2 -i"D:/Applications/MATLAB7/simulink/include" -i"D:
[rt_sim.c] "c:\ticcs\c6000\cgtools\bin\cl6x" -q -o2 -i"D:/Applications/MATLAB7/simulink/include" -i"D:/Appli
[c6416dsksurveil_hsrtdxcfg.s62] "c:\ticcs\c6000\cgtools\bin\cl6x" -q -o2 -i"D:/Applications/MATLAB7/simulink
[c6416dsksurveil_hsrtdxcfg_c.c] "c:\ticcs\c6000\cgtools\bin\cl6x" -q -o2 -i"D:/Applications/MATLAB7/simulink
[Linking...] "c:\ticcs\c6000\cgtools\bin\cl6x" -@"Custom_MW.lkf"

Build Complete,
  0 Errors, 0 Warnings, 0 Remarks.
```

Loading Program

Loading Program

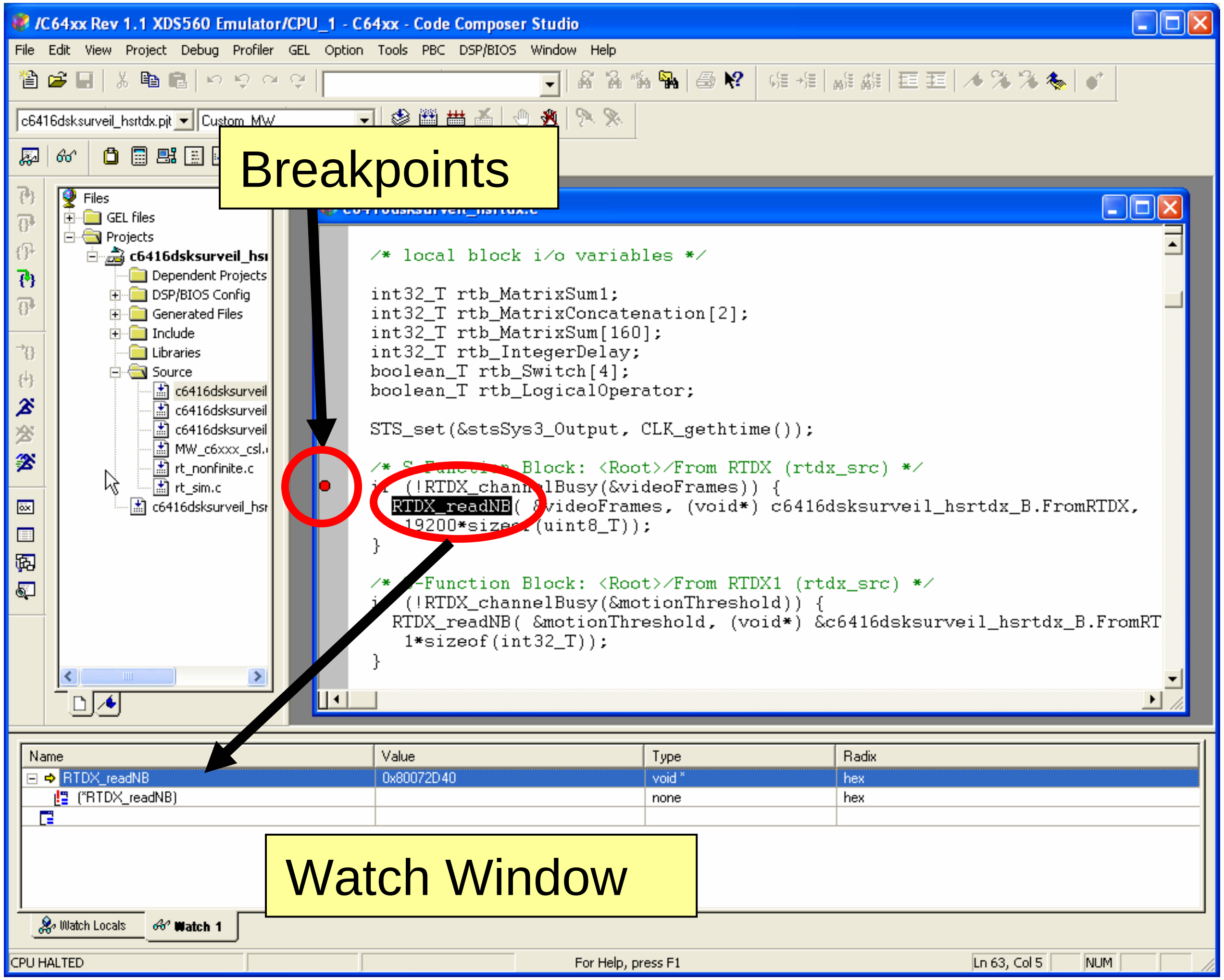
D:\Applications\MATLAB7\work\c6416dsksurveil_hsrtdx_

Loading sections.

.stack: 32752 of 262144 at 0x80031d40

2. Project Automation

- Utilize CCS to:
  - debug, test, and verify code
  - add and customize code



```
/* local block i/o variables */

int32_T rtb_MatrixSum1;
int32_T rtb_MatrixConcatenation[2];
int32_T rtb_MatrixSum[160];
int32_T rtb_IntegerDelay;
boolean_T rtb_Switch[4];
boolean_T rtb_LogicalOperator;

STS_set(&stsSys3_Output, CLK_gethtime());

/* S-Function Block: <Root>/From RTDX (rtdx_src) */
if (!RTDX_channelBusy(&videoFrames)) {
  RTDX_readNB( &videoFrames, (void*) c6416dsksurveil_hsrtdx_B.FromRTDX,
   19200*sizeof(uint8_T));
}

/* S-Function Block: <Root>/From RTDX1 (rtdx_src) */
if (!RTDX_channelBusy(&motionThreshold)) {
  RTDX_readNB( &motionThreshold, (void*) &c6416dsksurveil_hsrtdx_B.FromRT
   1*sizeof(int32_T));
}
```

Name	Value	Type	Radix
RTDX_readNB	0x80072D40	void *	hex
(*RTDX_readNB)		none	hex

3. Rapid Prototyping – Scenarios

Use Supported DSK/EVM#	***Use Custom Board + Emulator***	***Use TI DSP Simulator***
1. PC + DSK/EVM 2. Emulator Optional 3. Connect through USB, Parallel, PCI, or JTAG ports	1. PC + Custom Board + Emulator 2. Connect through JTAG port	1. PC 2. Simulator: Very Slow
1. Generate code 2. H-I-L prototyping* 3. RTDX 4. Access hw devices – ADC, DAC, daughter cards, etc.	1. Generate code 2. H-I-L prototyping* 3. RTDX	1. Generate code 2. S-I-L prototyping*
• Test and verify in CCS • Test and verify using MATLAB and Link for CCS	• Test and verify in CCS • Test and verify using MATLAB and Link for CCS	• Test and verify in CCS • Test and verify using MATLAB and Link for CCS

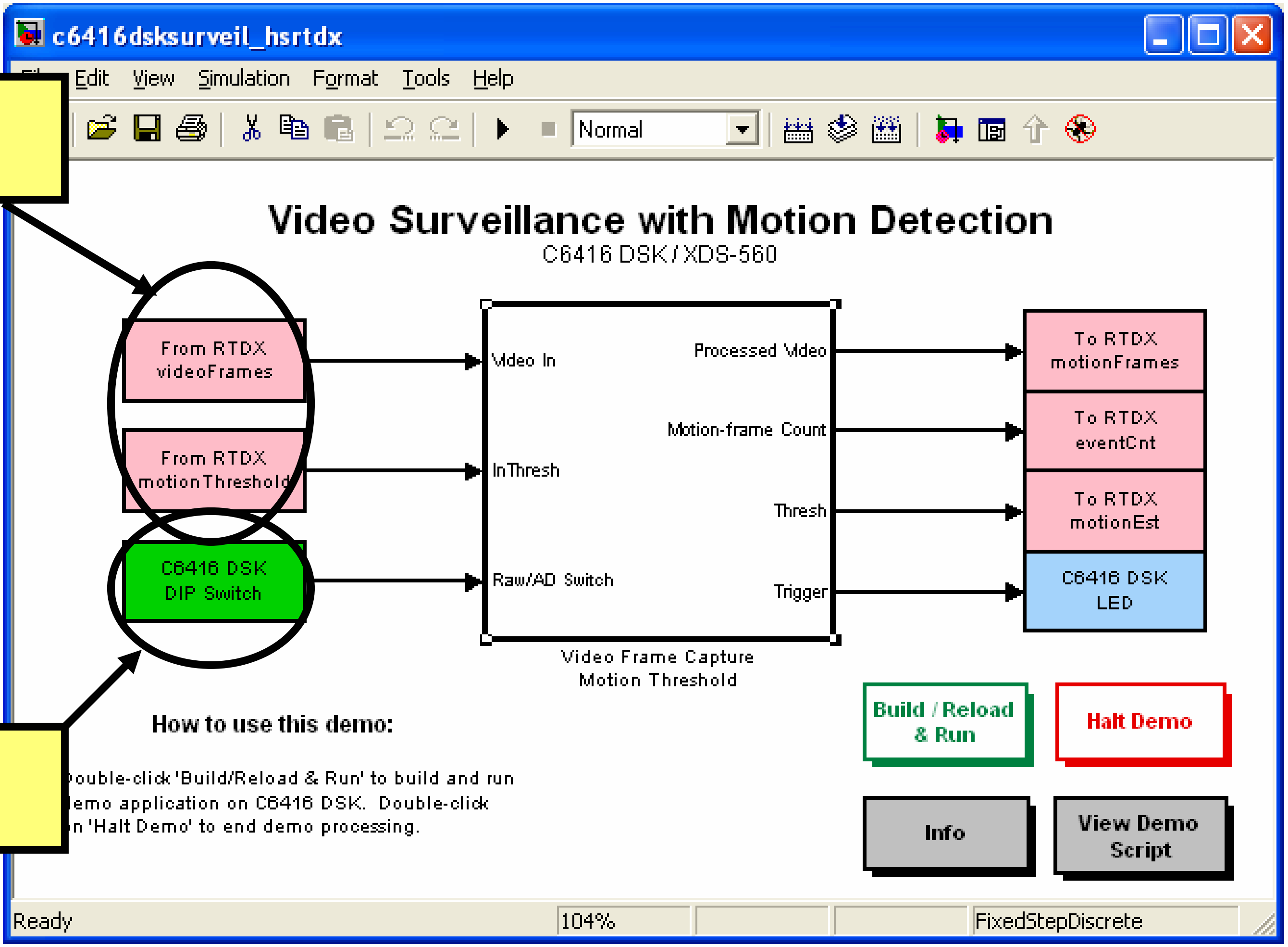
# Supports 6701 EVM, 6711, 6713, and 6416 DSKs. In v.2.1, DM642 EVM will also be supported.

*** From MATLAB**

Steps to Target the TI C6416 DSK

RTDX blocks

DIP Switch



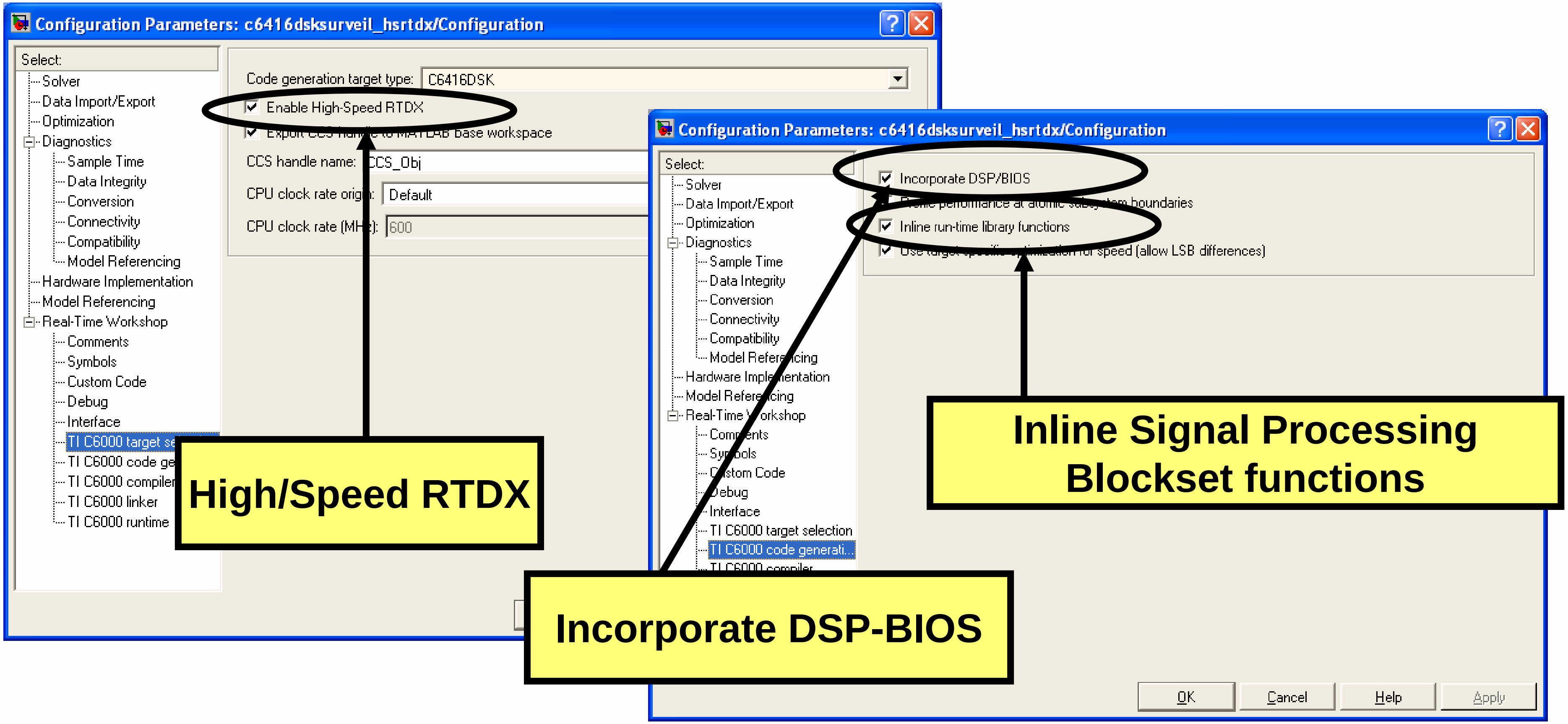
How to use this demo:

ouble-click 'Build/Reload & Run' to build and run
emo application on C6416 DSK. Double-click
n 'Halt Demo' to end demo processing.

```
>> c6416dsksurveil_hsrtdx
```

Steps to Target the TI C6416 DSK – 2

- Select target options (DSP/BIOS, compiler settings, etc)



Configuration Parameters: c6416dsksurveil_hsrtdx/Configuration

Select:
- Solver
- Data Import/Export
- Optimization
- Diagnostics
  - Sample Time
  - Data Integrity
  - Conversion
  - Connectivity
  - Compatibility
  - Model Referencing
- Hardware Implementation
- Model Referencing
- Real-Time Workshop
  - Comments
  - Symbols
  - Custom Code
  - Debug
  - Interface
  - TI C6000 target se
  - TI C6000 code ge
  - TI C6000 compiler
  - TI C6000 linker
  - TI C6000 runtime

Code generation target type: C6416DSK

☑ Enable High-Speed RTDX

☑ Export CCS handle to MATLAB base workspace

CCS handle name: CCS_Obj

CPU clock rate orig: Default

CPU clock rate (MHz): 600

Configuration Parameters: c6416dsksurveil_hsrtdx/Configuration

Select:
- Solver
- Data Import/Export
- Optimization
- Diagnostics
  - Sample Time
  - Data Integrity
  - Conversion
  - Connectivity
  - Compatibility
  - Model Referencing
- Hardware Implementation
- Model Referencing
- Real-Time Workshop
  - Comments
  - Symbols
  - Custom Code
  - Debug
  - Interface
  - TI C6000 target selection
  - TI C6000 code generati...
  - TI C6000 compiler

☑ Incorporate DSP/BIOS

☑ Profile performance at atomic subsystem boundaries

☑ Inline run-time library functions

☑ Use target specific optimization for speed (allow LSB differences)

High/Speed RTDX

Inline Signal Processing Blockset functions

Incorporate DSP-BIOS

Steps to Target the TI C6416 DSK – 3

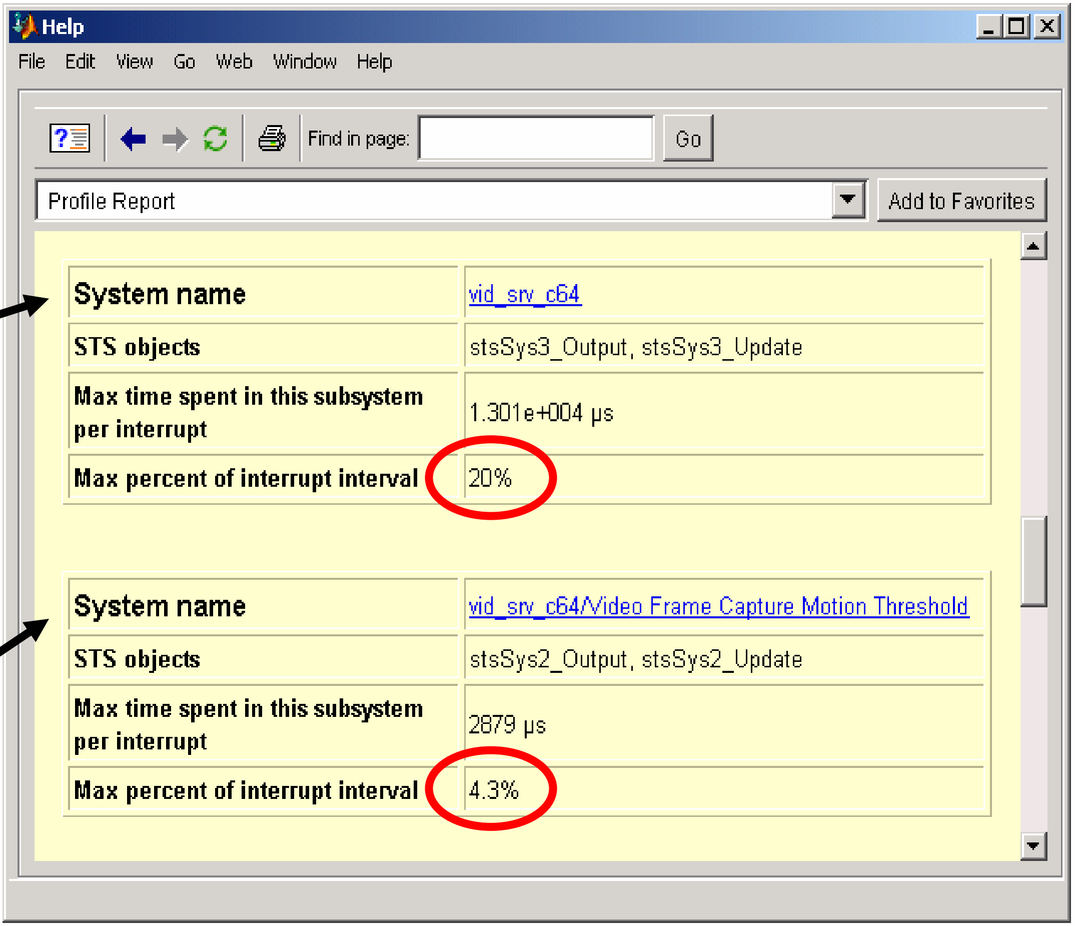
- Build process
  - Auto-generate ANSI C and ASM code
  - Integration of RTOS and scheduler
  - Create full CCS project in IDE Invoke compiler, linker, and download code
  - Run target

Target the TI C6416 DSK – 4

- Automatic profiling of program executing on DSP

System profiling
Includes entire DSP application code

Subsystem profiling



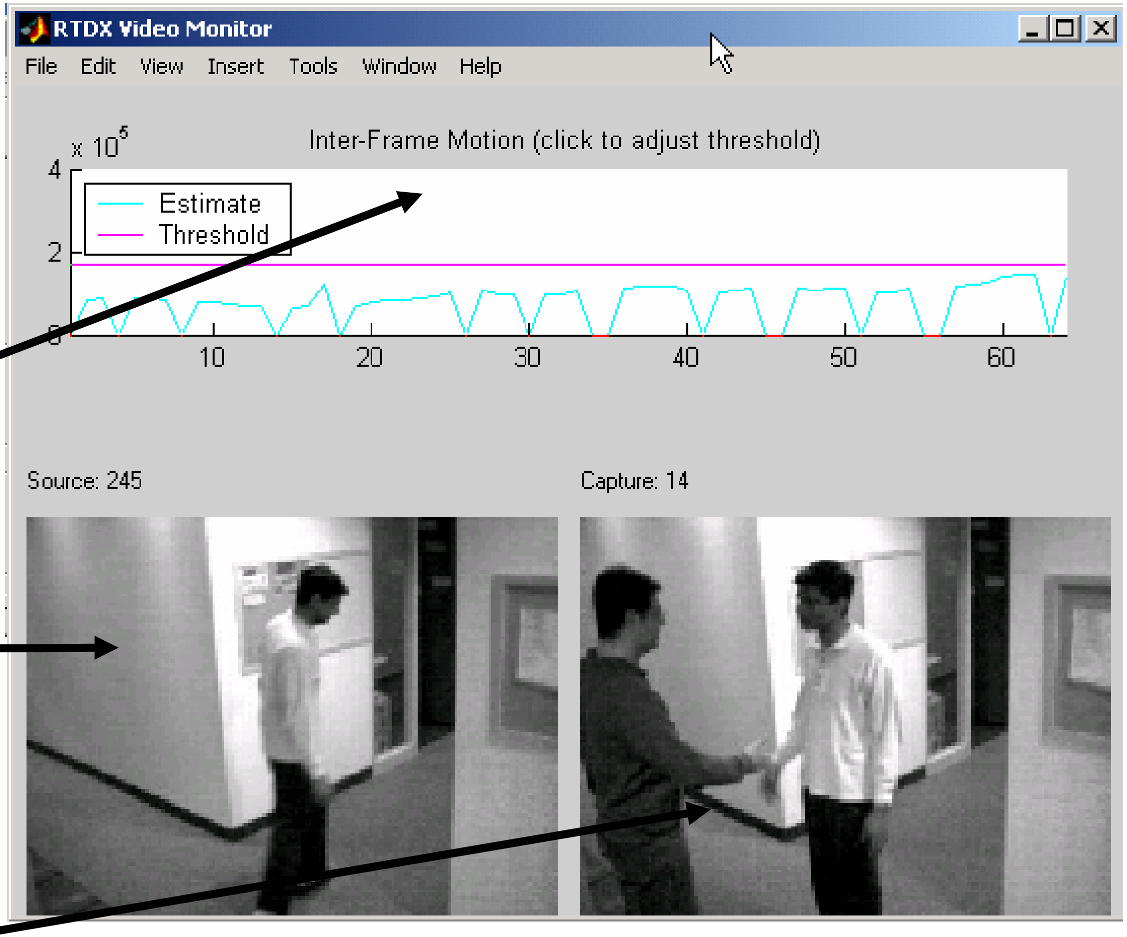
System name	vid_srv_c64
STS objects	stsSys3_Output, stsSys3_Update
Max time spent in this subsystem per interrupt	1.301e+004 µs
Max percent of interrupt interval	20%

System name	vid_srv_c64/Video Frame Capture Motion Threshold
STS objects	stsSys2_Output, stsSys2_Update
Max time spent in this subsystem per interrupt	2879 µs
Max percent of interrupt interval	4.3%

```
>> cc=ccsdsp;
>> profile(cc,'report')
```

Design Verification: Real-time Visualization

- Host-side visualization using Link for Code Composer Studio


RTDX Video Monitor
File Edit View Insert Tools Window Help
Inter-Frame Motion (click to adjust threshold)
x 10^5
4
2
0
10
20
30
40
50
60
Estimate
Threshold
Source: 245
Capture: 14

Log and plot estimates over time (scrolling data)

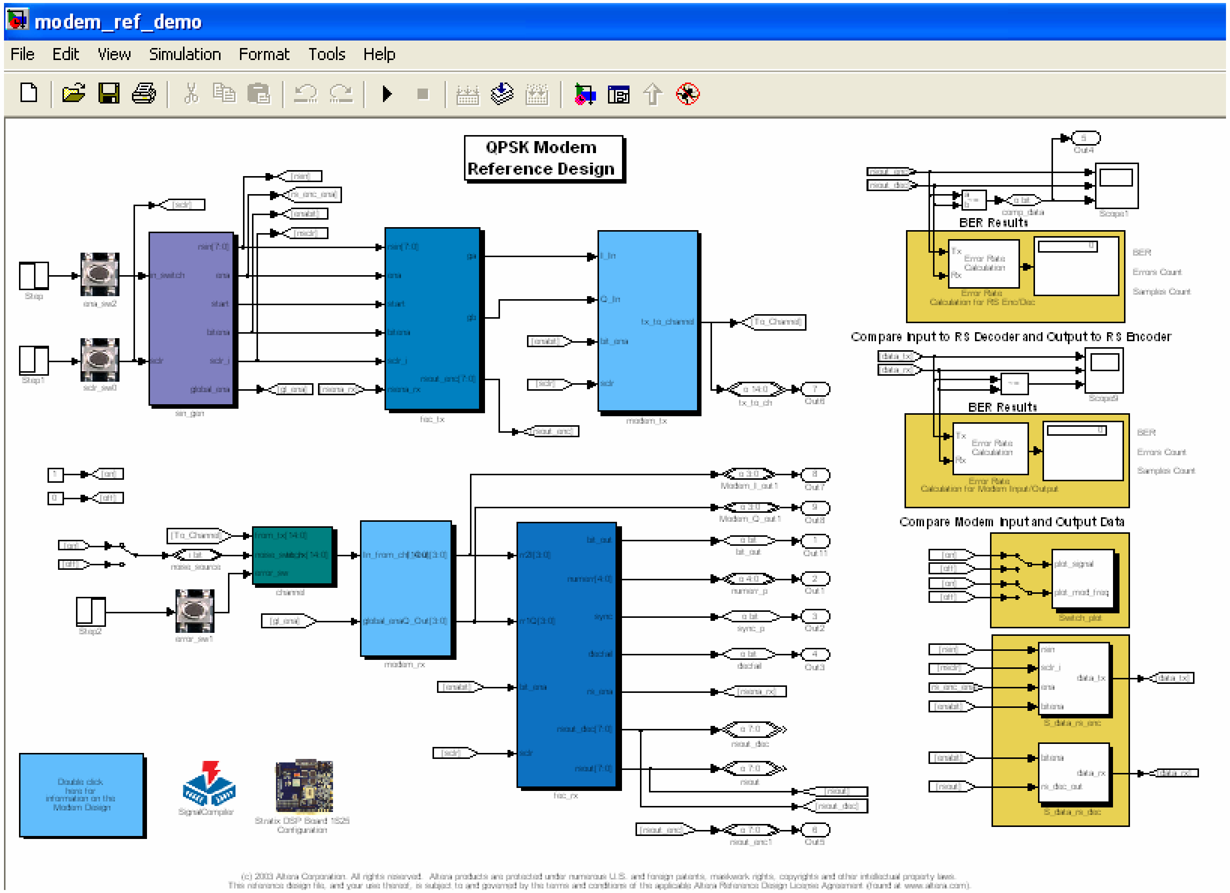
Input video frames

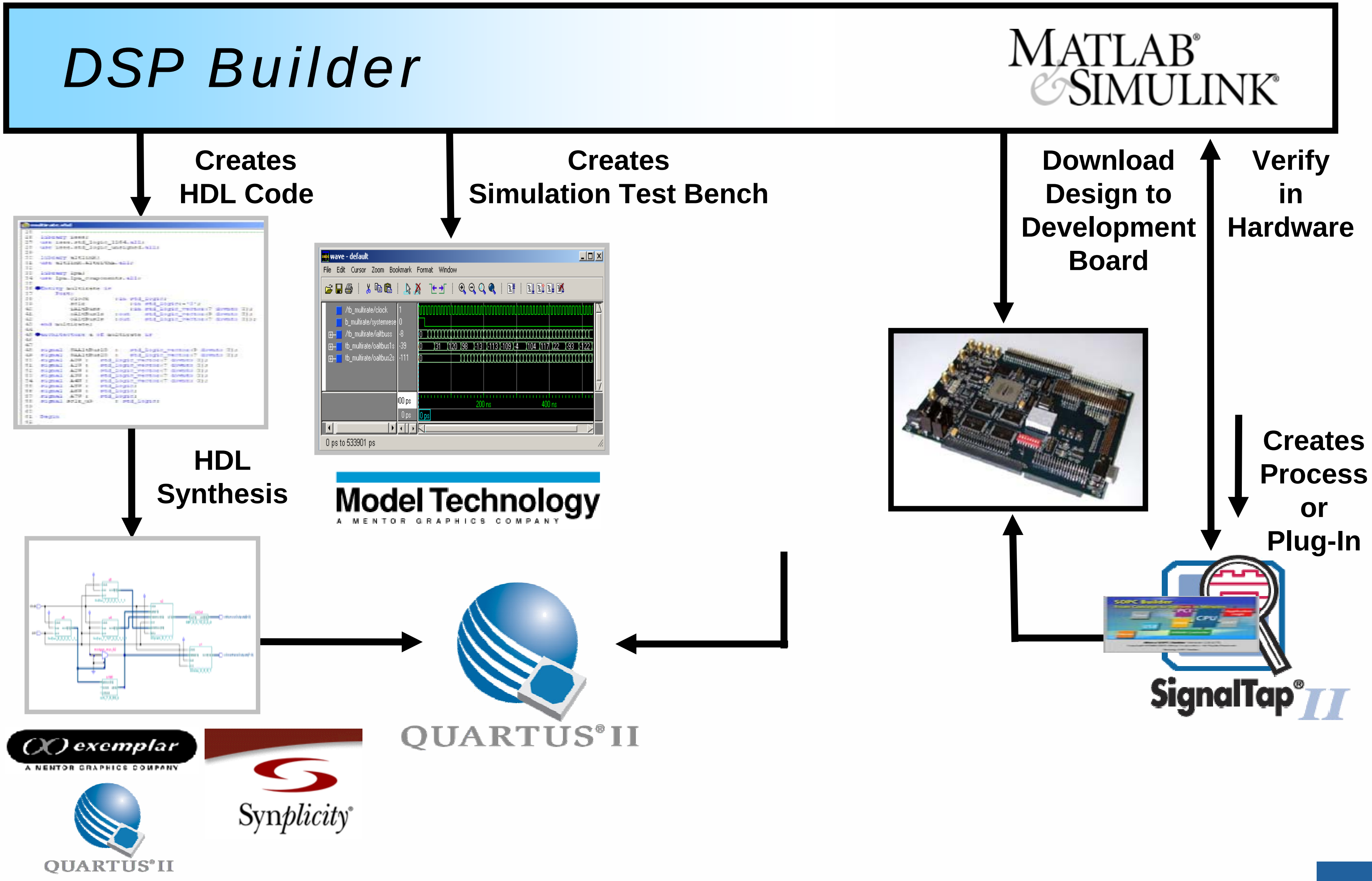
Captured frames

```
>> c6416dsksurveil_hsrtdx
```

Embedding Signal Processing Systems in Altera FPGAs Using the DSP Builder

Altera DSP Builder





DSP Builder

Creates HDL Code

Creates Simulation Test Bench

Download Design to Development Board

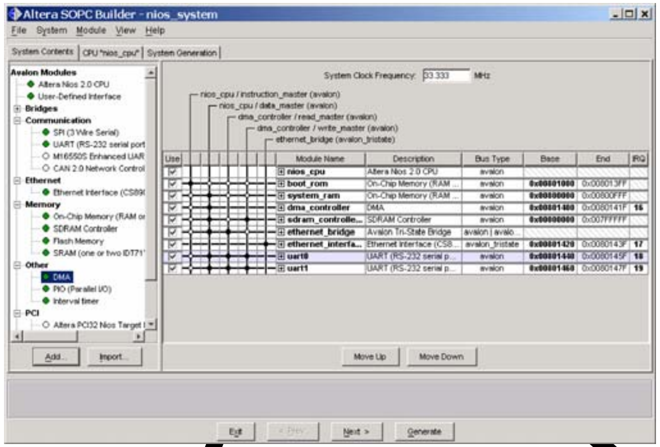
Verify in Hardware

HDL Synthesis

Creates Process or Plug-In

Custom Co-Processor Development

SOPC Builder

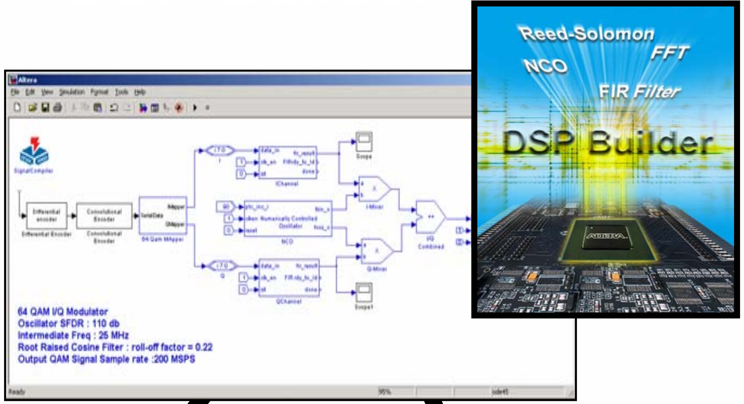


System Clock Frequency: 33.333 MHz

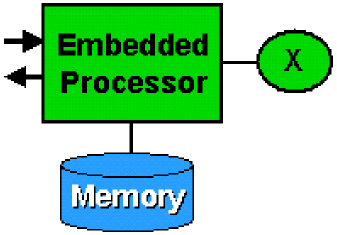
nios_cpu / instruction_master (avalon)
nios_cpu / data_master (avalon)
dma_controller / read_master (avalon)
dma_controller / write_master (avalon)
ethernet_bridge (avalon_tristate)

Use	Module Name	Description	Bus Type	Base	End	IRQ
☑	nios_cpu	Altera Nios 2.0 CPU	avalon			
☑	boot_rom	On-Chip Memory (RAM ...	avalon	0x00001000	0x000013FF	
☑	system_ram	On-Chip Memory (RAM ...	avalon	0x00000000	0x00800FFF	
☑	dma_controller	DMA	avalon	0x00001400	0x0080141F	16
☑	sdram_controlle...	SDRAM Controller	avalon	0x00000000	0x007FFFFF	
☑	ethernet_bridge	Avalon Tri-State Bridge	avalon | avalo...			
☑	ethernet_interfa...	Ethernet Interface (CS8...	avalon_tristate	0x00001420	0x0080143F	17
☑	uart0	UART (RS-232 serial p...	avalon	0x00001440	0x0080145F	18
☑	uart1	UART (RS-232 serial p...	avalon	0x00001460	0x0080147F	19

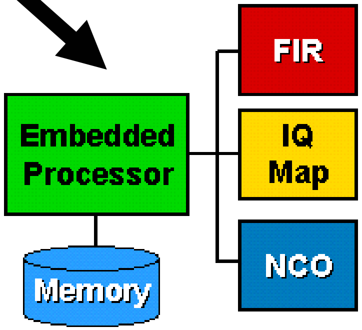
DSP Builder



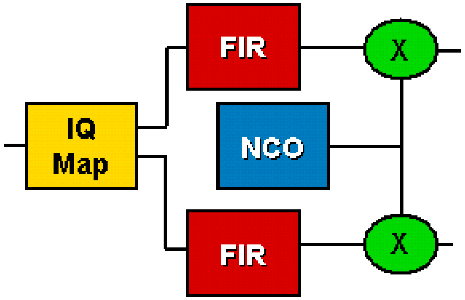
64 QAM I/Q Modulator
Oscillator SFDR : 110 db
Intermediate Freq : 25 MHz
Root Raised Cosine Filter : roll-off factor = 0.22
Output QAM Signal Sample rate :200 MSPS



Stand-Alone Processor



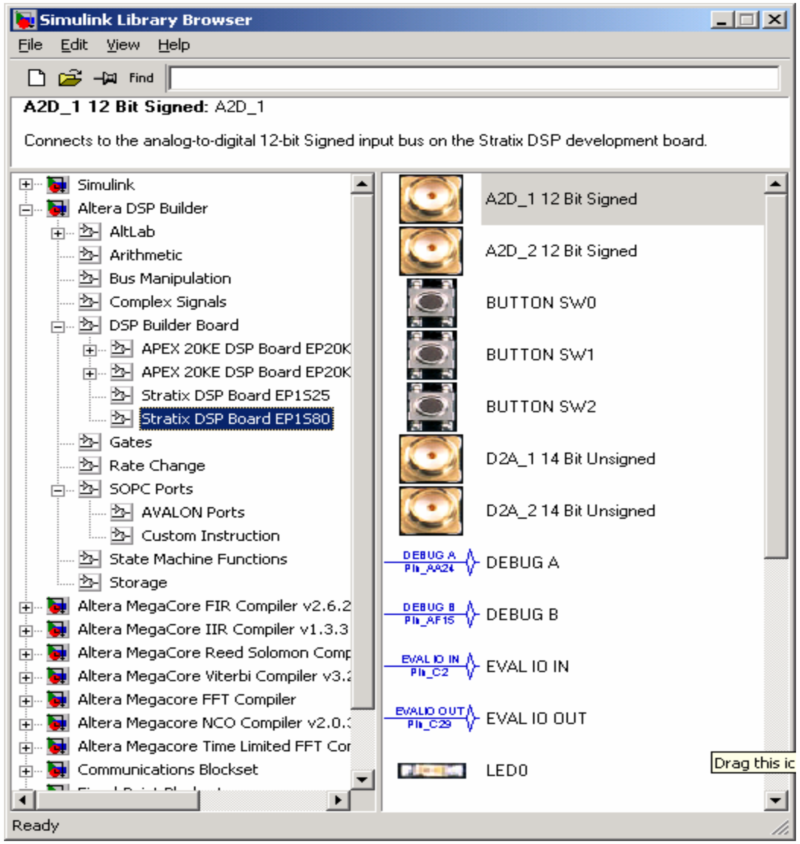
Processor + Co-Processor



Dedicated Hardware Architecture

DSP Builder Library Components

- Arithmetic
- Bus Manipulation
- Complex Signals
- Logical Components
- SOPC Ports
- Storage
- MegaCore® IP
- Rate Change
- State Machine
- Altera Library
- DSP Board



A2D_1 12 Bit Signed: A2D_1

Connects to the analog-to-digital 12-bit Signed input bus on the Stratix DSP development board.

- Simulink
- Altera DSP Builder
  - AltLab
  - Arithmetic
  - Bus Manipulation
  - Complex Signals
  - DSP Builder Board
    - APEX 20KE DSP Board EP20K
    - APEX 20KE DSP Board EP20K
    - Stratix DSP Board EP1S25
    - Stratix DSP Board EP1S80
  - Gates
  - Rate Change
  - SOPC Ports
    - AVALON Ports
    - Custom Instruction
  - State Machine Functions
  - Storage
- Altera MegaCore FIR Compiler v2.6.2
- Altera MegaCore IIR Compiler v1.3.3
- Altera MegaCore Reed Solomon Comp
- Altera MegaCore Viterbi Compiler v3.2
- Altera Megacore FFT Compiler
- Altera Megacore NCO Compiler v2.0.3
- Altera Megacore Time Limited FFT Cor
- Communications Blockset

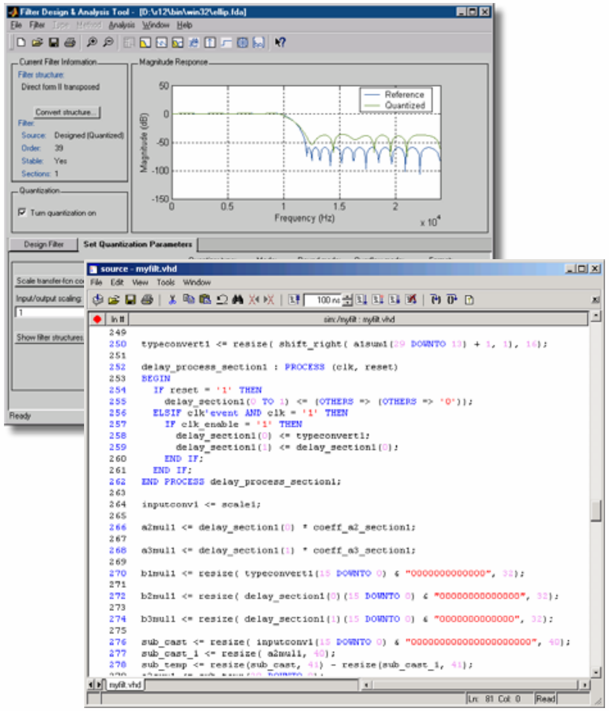
A2D_1 12 Bit Signed
A2D_2 12 Bit Signed
BUTTON SW0
BUTTON SW1
BUTTON SW2
D2A_1 14 Bit Unsigned
D2A_2 14 Bit Unsigned
DEBUG A
DEBUG B
EVAL IO IN
EVAL IO OUT
LED0

Other MathWorks Products for HDL Code Generation and Verification

Filter Design HDL Coder for Digital Filters

- Description
  - Design IIR fixed-point filter
  - Generate synthesizable VHDL or Verilog
  - Verify implementation through co-simulation

- What you will see
  - Fixed-point filter design
  - Automatic HDL generation with Filter Design HDL Coder
  - Verify implementation with Link for ModelSim



Filter structure: Direct form II transposed

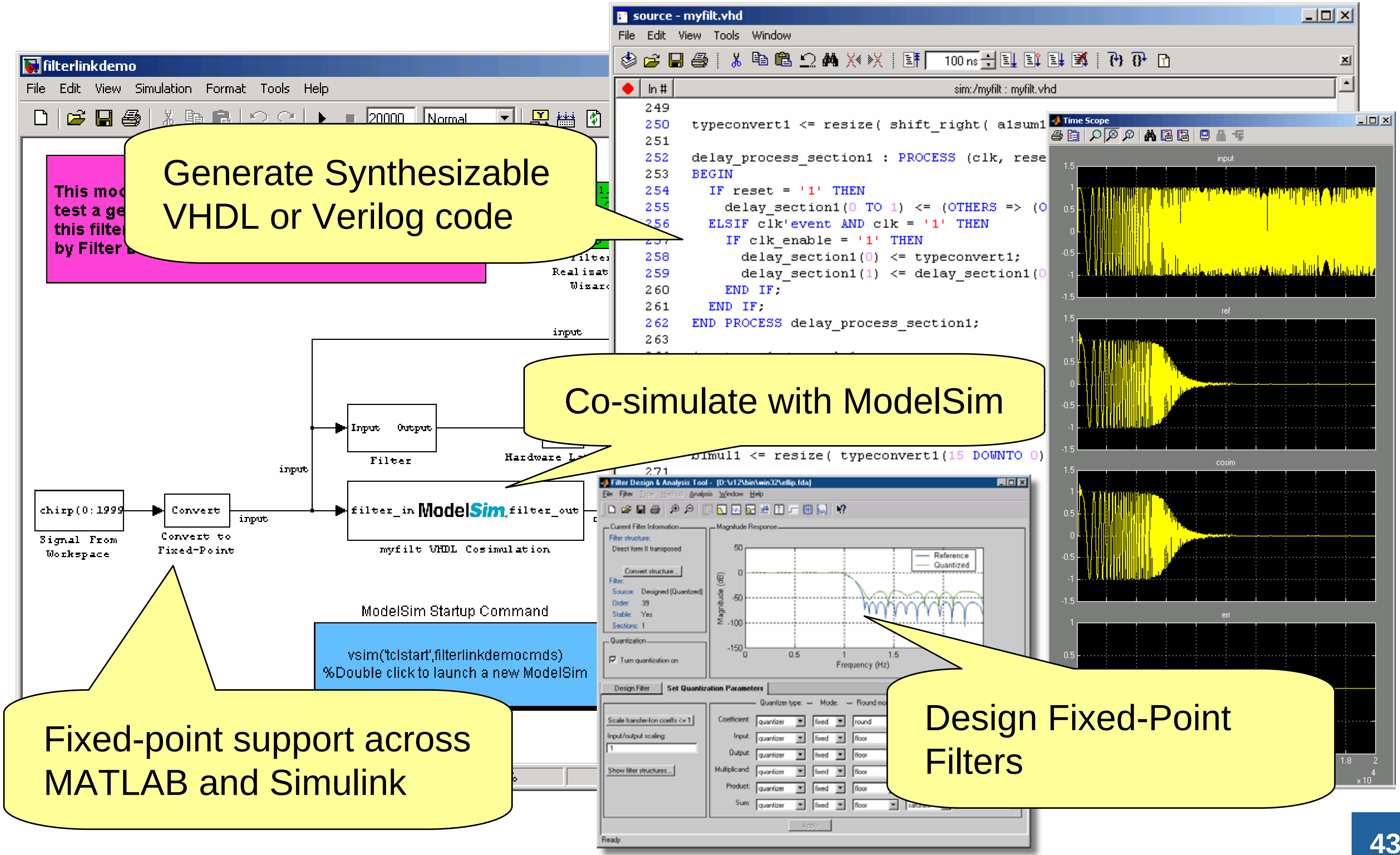
Filter:
Source: Designed (Quantized)
Order: 39
Stable: Yes
Sections: 1

☑ Turn quantization on

Input/output scaling: 1

```

typeconvert1 <= resize( shift_right( a1sum1(29 DOWNTO 13) + 1, 1), 16);

delay_process_section1 : PROCESS (clk, reset)
BEGIN
  IF reset = '1' THEN
    delay_section1(0 TO 1) <= (OTHERS => (OTHERS => '0'));
  ELSIF clk'event AND clk = '1' THEN
    IF clk_enable = '1' THEN
      delay_section1(0) <= typeconvert1;
      delay_section1(1) <= delay_section1(0);
    END IF;
  END IF;
END PROCESS delay_process_section1;

inputconv1 <= scale1;

a2mul1 <= delay_section1(0) * coeff_a2_section1;

a3mul1 <= delay_section1(1) * coeff_a3_section1;

b1mul1 <= resize( typeconvert1(15 DOWNTO 0) & "0000000000000", 32);

b2mul1 <= resize( delay_section1(0)(15 DOWNTO 0) & "00000000000000", 32);

b3mul1 <= resize( delay_section1(1)(15 DOWNTO 0) & "0000000000000", 32);

sub_cast <= resize( inputconv1(15 DOWNTO 0) & "0000000000000000000000", 40);
sub_cast_1 <= resize( a2mul1, 40);
sub_temp <= resize(sub_cast, 41) - resize(sub_cast_1, 41);
```

Link for ModelSim



Generate Synthesizable VHDL or Verilog code

Co-simulate with ModelSim

Fixed-point support across MATLAB and Simulink

Design Fixed-Point Filters

source - myfilt.vhd — sim:/myfilt : myfilt.vhd

```
typeconvert1 <= resize( shift_right( a1sum1

delay_process_section1 : PROCESS (clk, rese
BEGIN
  IF reset = '1' THEN
    delay_section1(0 TO 1) <= (OTHERS => (O
  ELSIF clk'event AND clk = '1' THEN
    IF clk_enable = '1' THEN
      delay_section1(0) <= typeconvert1;
      delay_section1(1) <= delay_section1(0
    END IF;
  END IF;
END PROCESS delay_process_section1;

b1mul1 <= resize( typeconvert1(15 DOWNTO 0)
```

ModelSim Startup Command

vsim('tclstart',filterlinkdemocmds)
%Double click to launch a new ModelSim

Filter Design & Analysis Tool - [D:\r12\bin\win32\ellip.fda]

Current Filter Information

Filter structure: Direct form II transposed

Filter: Source: Designed (Quantized); Order: 39; Stable: Yes; Sections: 1

Quantization: ☑ Turn quantization on

Set Quantization Parameters

Input/output scaling: 1

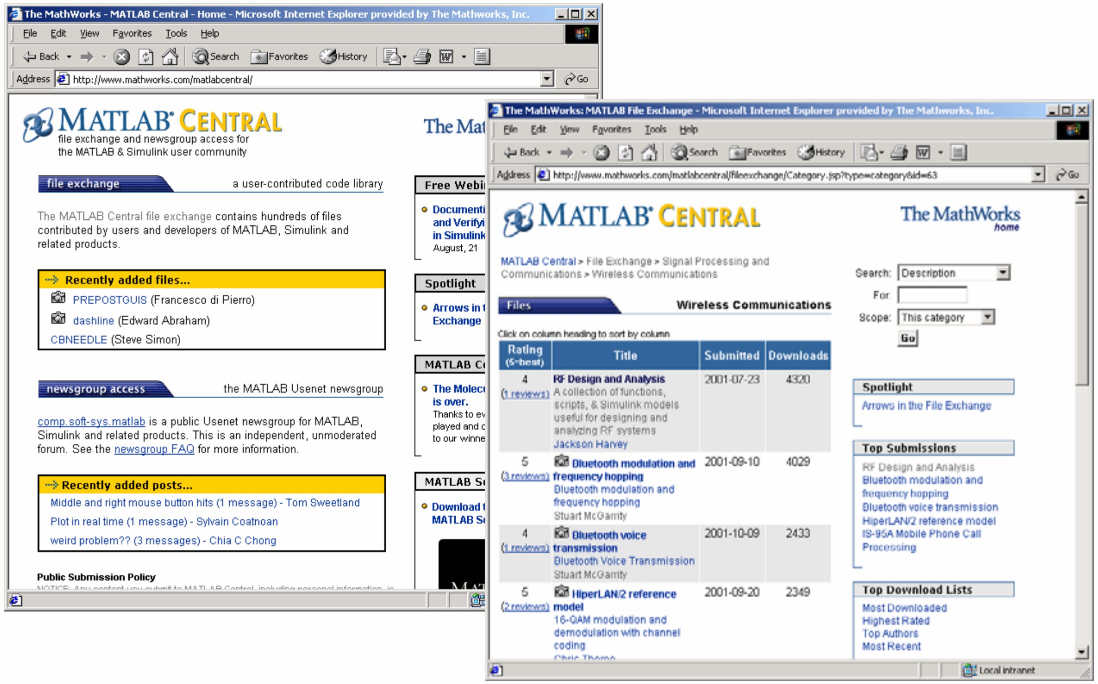
	Quantizer type:	Mode:	Round mo
Coefficient:	quantizer	fixed	round
Input:	quantizer	fixed	floor
Output:	quantizer	fixed	floor
Multiplicand:	quantizer	fixed	floor
Product:	quantizer	fixed	floor
Sum:	quantizer	fixed	floor

Simulink and Model-Based Design Produce Results Across Industries

TOYOTA	Standard for Powertrain Controls Production Code Development
LOCKHEED MARTIN	JSF Flight Control System
zyray WIRELESS	W-CDMA Baseband Processors
TEXAS INSTRUMENTS	Specialty Chipsets for DSP Customers

MATLAB Central

- www.matlabcentral.com
- Over 1,500 MathWorks- and user-contributed files
- MATLAB files and Simulink models for download



The MathWorks - MATLAB Central - Home - Microsoft Internet Explorer provided by The Mathworks, Inc.

File Edit View Favorites Tools Help
Back Search Favorites History
Address
Go

http://www.mathworks.com/matlabcentral/

MATLAB® CENTRAL
file exchange and newsgroup access for
the MATLAB & Simulink user community

file exchange a user-contributed code library

The MATLAB Central file exchange contains hundreds of files contributed by users and developers of MATLAB, Simulink and related products.

Recently added files...
- PREPOSTGUIS (Francesco di Pierro)
- dashline (Edward Abraham)
- CBNEEDLE (Steve Simon)

newsgroup access the MATLAB Usenet newsgroup

comp.soft-sys.matlab is a public Usenet newsgroup for MATLAB, Simulink and related products. This is an independent, unmoderated forum. See the newsgroup FAQ for more information.

Recently added posts...
- Middle and right mouse button hits (1 message) - Tom Sweetland
- Plot in real time (1 message) - Sylvain Coatnoan
- weird problem?? (3 messages) - Chia C Chong

Public Submission Policy

The Mat

Free Webi
- Documenti and Verifyi in Simulink August, 21

Spotlight
- Arrows in t Exchange

MATLAB C
- The Molecu is over. Thanks to ev played and c to our winne

MATLAB S
- Download t MATLAB S

The MathWorks: MATLAB File Exchange - Microsoft Internet Explorer provided by The Mathworks, Inc.

File Edit View Favorites Tools Help
Back Search Favorites History
Address
Go

http://www.mathworks.com/matlabcentral/fileexchange/Category.jsp?type=category&id=63

MATLAB® CENTRAL

The MathWorks *home*

MATLAB Central > File Exchange > Signal Processing and Communications > Wireless Communications

Search: Description
For:
Scope: This category
Go

Files **Wireless Communications**

Click on column heading to sort by column

Rating (5=best)	Title	Submitted	Downloads
4 (1 reviews)	**RF Design and Analysis** A collection of functions, scripts, & Simulink models useful for designing and analyzing RF systems Jackson Harvey	2001-07-23	4320
5 (3 reviews)	**Bluetooth modulation and frequency hopping** Bluetooth modulation and frequency hopping Stuart McGarrity	2001-09-10	4029
4 (1 reviews)	**Bluetooth voice transmission** Bluetooth Voice Transmission Stuart McGarrity	2001-10-09	2433
5 (2 reviews)	**HiperLAN/2 reference model** 16-QAM modulation and demodulation with channel coding Chris Thomo	2001-09-20	2349

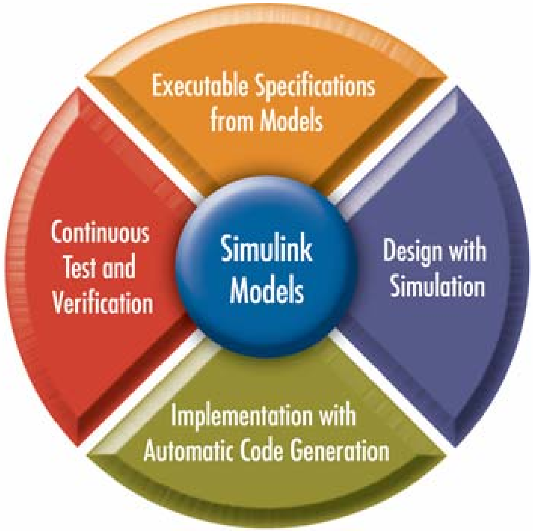
Spotlight
Arrows in the File Exchange

Top Submissions
- RF Design and Analysis
- Bluetooth modulation and frequency hopping
- Bluetooth voice transmission
- HiperLAN/2 reference model
- IS-95A Mobile Phone Call Processing

Top Download Lists
- Most Downloaded
- Highest Rated
- Top Authors
- Most Recent

Summary

- Simulink brings:
  - Model-Based Design to large-scale projects
  - More comprehensive coverage of embedded system development
  - New domains and applications
- Visit the web for more
  www.mathworks.com/r14
- Visit us at the booth to see more product demonstrations


Executable Specifications from Models
Continuous Test and Verification
Simulink Models
Design with Simulation
Implementation with Automatic Code Generation