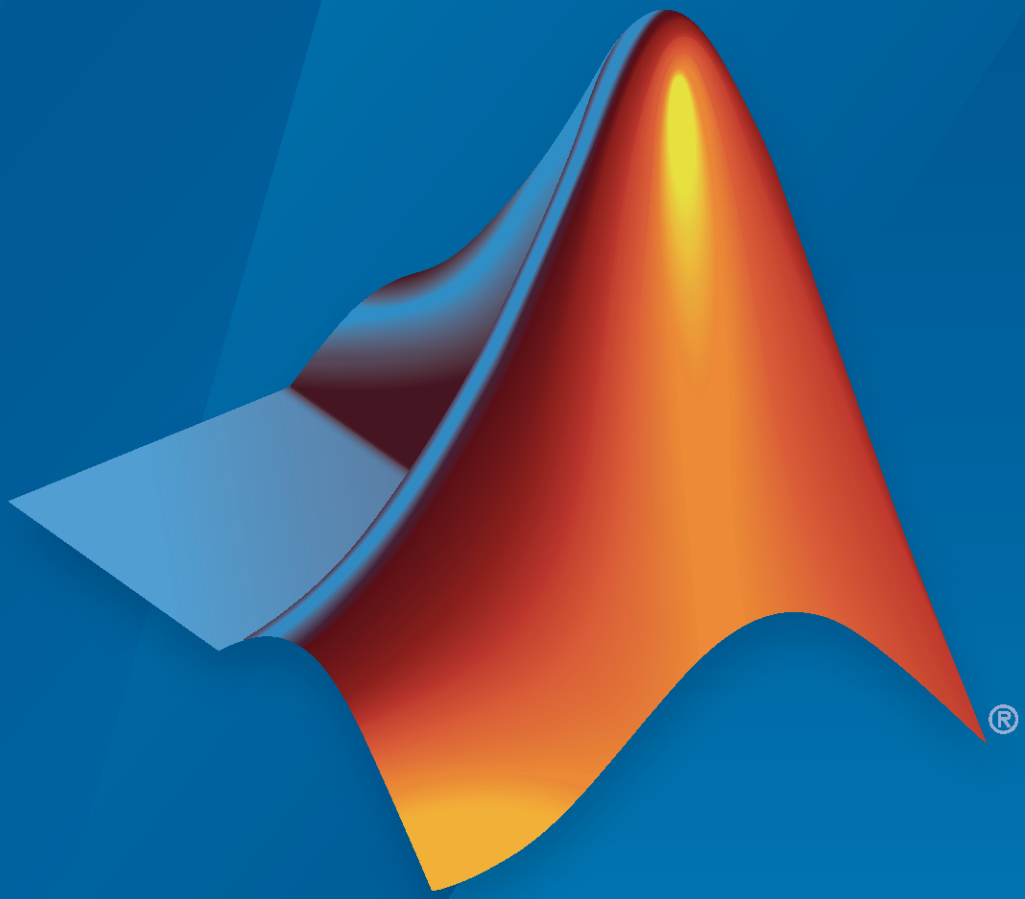
SoC Blockset™ Release Notes



MATLAB® & SIMULINK®

MathWorks®

How to Contact MathWorks

Latest news: `www.mathworks.com`

Sales and services: `www.mathworks.com/sales_and_services`

User community: `www.mathworks.com/matlabcentral`

Technical support: `www.mathworks.com/support/contact_us`

Phone: 508-647-7000

The MathWorks, Inc.
1 Apple Hill Drive
Natick, MA 01760-2098

SoC Blockset™ Release Notes

Contents

R2023a

R2022b

R2022a

R2021b

R2021a

R2020b

R2020a

R2019b

R2019a

R2023a

AXI4 Random Access Memory block adds memory initialization and logging

You can now load an initial value or log memory content when using the AXI4 Random Access Memory block.

- To initialize the memory, open the block mask and select **Initialize memory content** on the **Main** tab. Then, specify the initial value in the **Initial value** box.
- To log the contents of the memory after simulation, open the block mask and select **Log memory content** on the **Main** tab.

For more information, see AXI4 Random Access Memory.

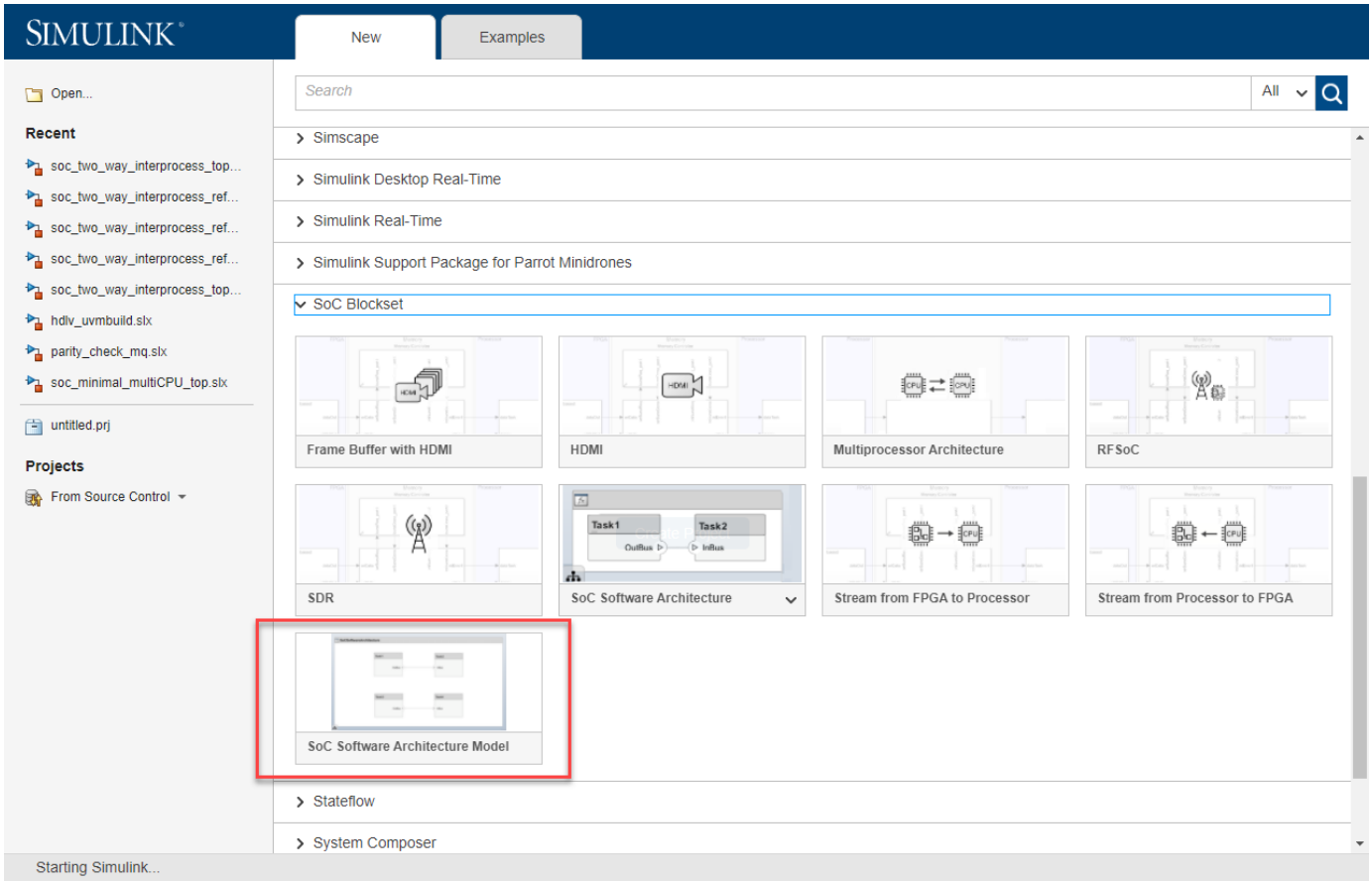
AXI-Stream protocol supports TLAST in simulation

Simulation of AXI-Stream protocols now supports the TLAST signal as an indicator of the last word in a data packet. This feature allows you to simulate data packets of variable sizes. In previous releases, TLAST was supported when targeting an FPGA or SoC device, and now you can use this feature in simulation. For more information about the AXI4-Stream protocol, see “AXI4-Stream Interface”.

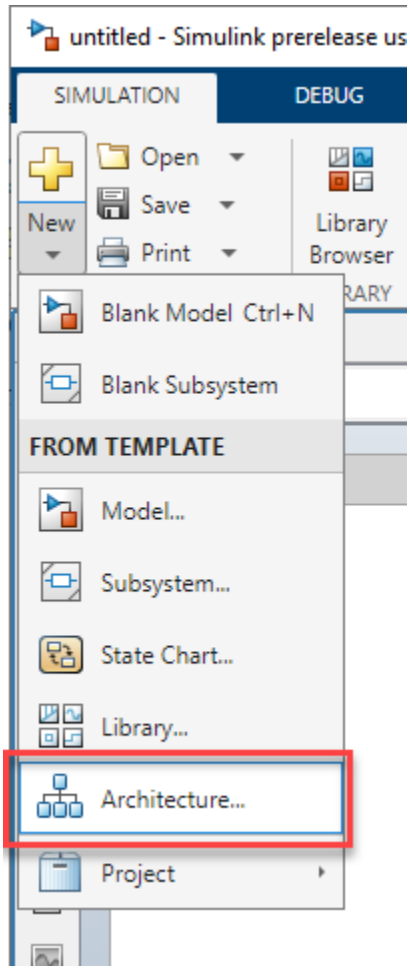
New SoC software architecture template

Model software architectures, analyze task schedules, and develop full system models. To open this template, select one of these options:

- On the Simulink® start page, scroll to the **SoC Blockset** section, and select **SoC Software Architecture Model**.



- Open Simulink and, on the **Simulation** tab, expand the **New** button and select **Architecture**. In the window that opens, select **SoC Software Architecture Model**.



For more information, see “SoC Software Architecture Template”.

Embedded Linux Explorer

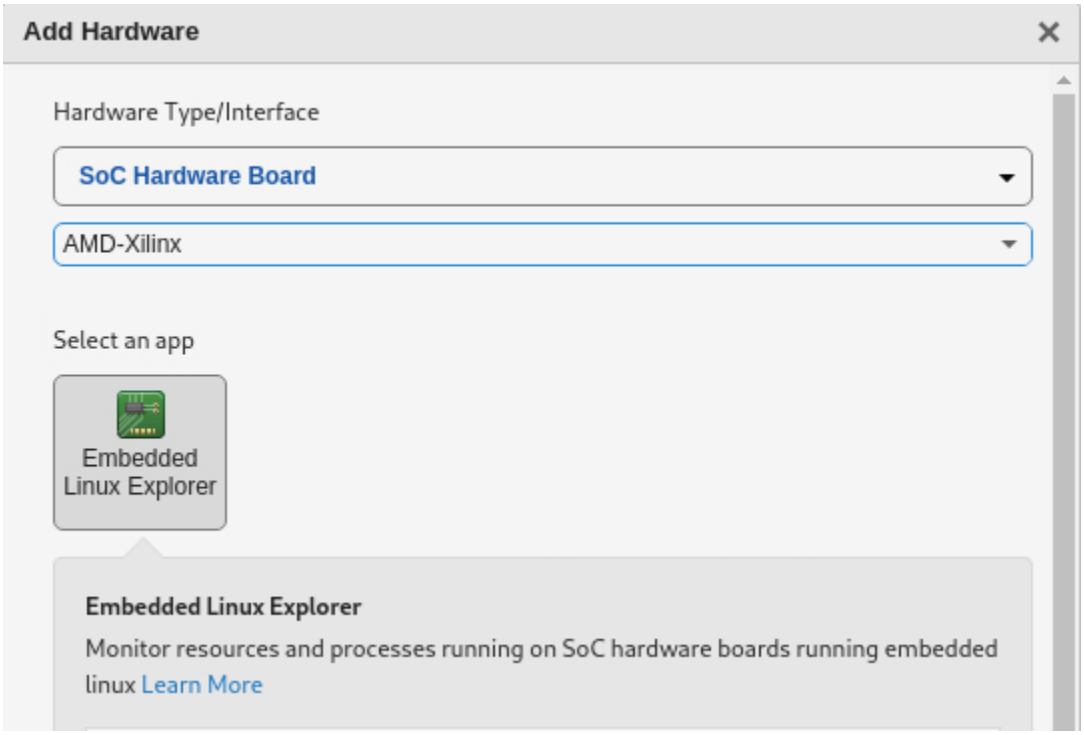
In R2023a, the new **Embedded Linux Explorer** app enables you to monitor resources and processes running on an SoC device.

To access the **Embedded Linux Explorer**:

- In the MATLAB® command line enter:

  ```
  embeddedLinuxExplorer
  ```

- Open the **Hardware Manager** and select **SoC Hardware Board** for **Hardware Type**. Then, set **Select your hardware manufacturer** to `AMD-Xilinx` or `Intel` and click the **Embedded Linux Explorer** button.



Import HDL IP core into Simulink model

You can now import an existing HDL IP core as a block into an SoC Simulink model. The new **HDL IP Importer** tool takes you through the steps to generate a library block. Integrate the generated block into your Simulink model. You can then simulate, build, and deploy the model on a hardware board using the **SoC Builder** tool.

To open the **HDL IP Importer** tool, at the MATLAB command prompt, enter `hdlIPImporter`.

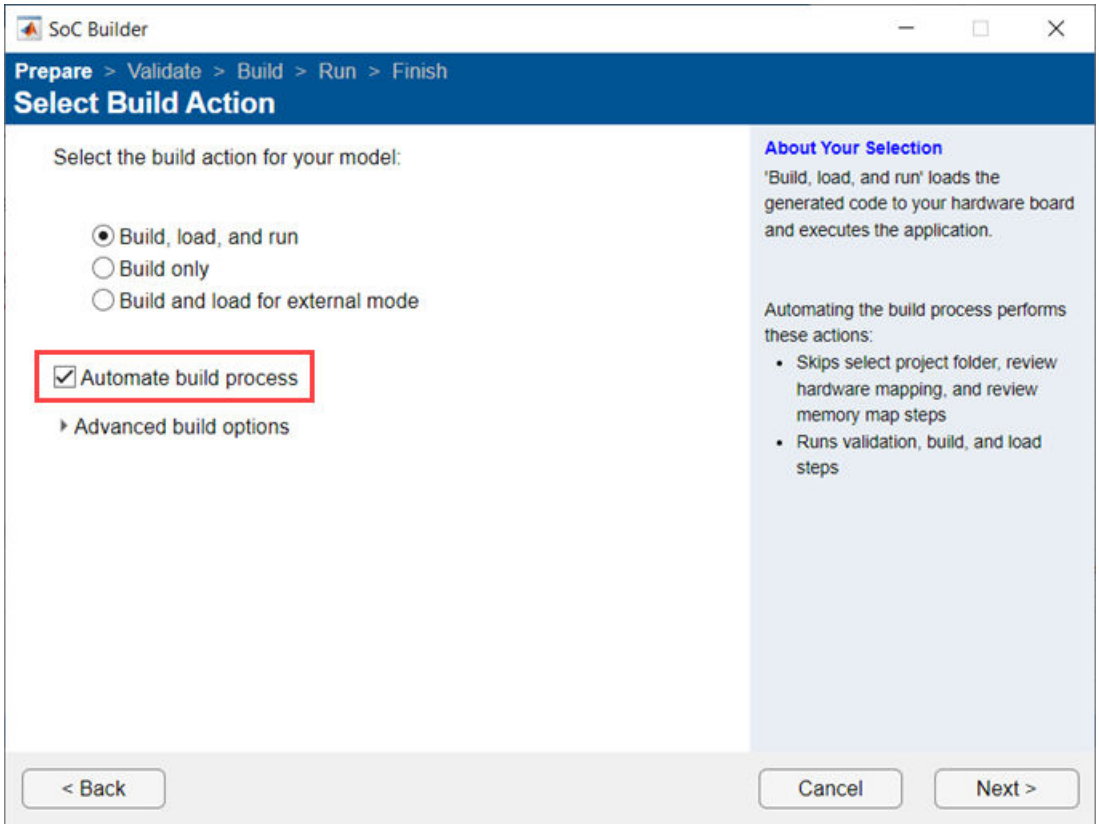
Run SoC Builder workflow using command line interface

You can now use the MATLAB commands to run the SoC Builder workflow. Previously, the workflow was graphical user interface (GUI) based only. Use the new `socModelBuilder` object to control and run the generation workflow. You can also use this object to load the existing binaries onto the hardware board.

Enhancements to SoC Builder tool

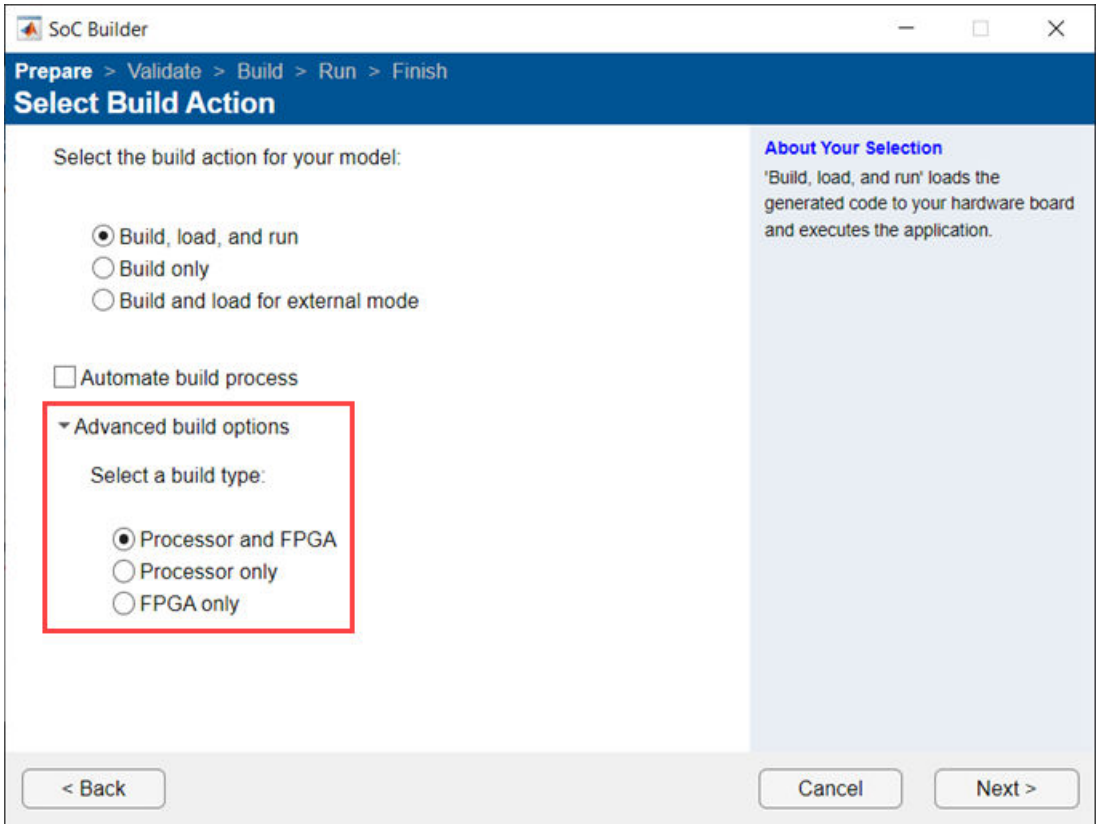
- **Reduced number of clicks** — To build the SoC model with fewer clicks, you can now automate the build process in the **SoC Builder** tool. Automating the build process performs these actions.
  - Skips these steps — Select project folder, review hardware mapping, and review memory map
  - Runs these steps — Validate, build, and load

  To automate the build process, on the **Select Build Action** screen, select **Automate build process**.



- **Advanced build options** — You can now select a build type as one of these options.
  - **Processor and FPGA** (default) — Build the processor and FPGA models in your top model.
  - **Processor only** — Build only the processor model in your top model.
  - **FPGA only** — Build only the FPGA model in your top model.

  To select these options, on the **Select Build Action** screen, expand **Advanced build options**.



SoC Builder workflow packages project files for sharing

The SoC Builder workflow now packages the project files into a ZIP archive. The packaged zip file contains all the files required to deploy the design onto the hardware board. Unzip this packaged ZIP archive to load and run your design from a different host machine.

SoC Builder workflow supports auto generation of host interface script

The SoC Builder workflow now generates the host interface script for your hardware design. The script contains the DUT ports and interface mapping information. Use this script to access the board memory, DUT registers, and AXI4-Stream interfaces from MATLAB. The generated MATLAB files are:

- `gs_modelName_setup` — This script adds the AXI4 slave, AXI4-Stream, and memory interfaces. The script also contains DUT port objects that contain the port name, direction, data type, and interface mapping information. The script then maps the DUT ports to the corresponding interfaces.
- `gs_modelName_interface` — This script creates a target object, instantiates the setup script `gs_modelName_setup`, and then connects to the target hardware. The script then sends read and write commands to the generated HDL IP core.

SoC Blockset Support Package for AMD-Xilinx Versal ACAP Devices: Design, analyze, and prototype for Versal ACAP Devices

SoC Blockset™ Support Package for AMD®-Xilinx® Versal® ACAP Devices enables you to model, simulate, analyze, and prototype hardware and software architectures on Xilinx Versal ACAP devices using SoC Blockset. The support package features key capabilities including software and hardware code generation (with required coder products) and software profiling.

For the complete documentation of this support package, see “AMD-Xilinx Versal ACAP Devices”. The documentation of this support package is included in the SoC Blockset product documentation.

SoC Blockset Support Package for Infineon AURIX Microcontrollers: Model-based embedded software development for Infineon AURIX TC4x microcontrollers

SoC Blockset Support Package for Infineon® AURIX™ Microcontrollers enables you to model, generate, and deploy embedded software on Infineon AURIX TC4x microcontrollers. The support package includes key capabilities like multicore modeling and targeting, generating code for I/O device drivers and ISR/Interrupt for TriCore, generating code for SIMD PPU core, verifying and validating the code using PIL, and monitoring and tuning the parameters.

For the complete documentation of this support package, see “Infineon AURIX Microcontrollers”. The documentation of this support package is included in the SoC Blockset product documentation.

Support for Versal AI Core Series VCK190 Evaluation Kit

Simulate SoC models and deploy them on the Xilinx Versal AI Core Series VCK190 Evaluation Kit. You can use this board to integrate your algorithms with a fixed reference design.

To use the VCK190 evaluation kit, open the **SoC Model Creator** and set **Reference design board** to `Xilinx Versal VCK190 Evaluation Kit`.

To use this feature, you must install the SoC Blockset Support Package for AMD-Xilinx Versal ACAP Devices.

Support for OTAVA DTRX2 mmWave radio card

Integrate the OTAVA DTRX2 mmWave radio card with the Xilinx Zynq® UltraScale+™ RFSoC ZCU208 Evaluation Kit by using the new OTAVA DTRX2 block in Simulink. The OTAVA DTRX2 block provides an RF front-end interface in the mmWave frequency band to implement your RF applications on the ZCU208 board.

For an example using the OTAVA DTRX2 block, see “Transmit and Receive Tone Using OTAVA DTRX2 mmWave Radio Card” (SoC Blockset Support Package for Xilinx Devices).

To use this feature, you must install the SoC Blockset Support Package for Xilinx Devices.

Support for Xilinx Aurora interface

The Aurora 64B66B block provides high-speed serial communication using the Xilinx Aurora 64B/66B cores.

For an example using the Aurora 64B66B block, see “Import Custom HDL IP into SoC Blockset Design” (SoC Blockset Support Package for Xilinx Devices).

To use this feature, you must install the SoC Blockset Support Package for Xilinx Devices.

Updates to supported software

The SoC Blockset product now supports these software versions. For a full list of supported software, see “Supported Third-Party Tools and Hardware”.

- Intel® Quartus® Prime Standard 21.1
- Xilinx Vivado® 2022.1

Functionality being removed or changed

SoC Blockset Support Package for Texas Instruments C2000 Processors has transitioned into C2000 Microcontroller Blockset
Behavior change

Starting in R2023a, SoC Blockset Support Package for Texas Instruments® C2000™ Processors has transitioned into C2000 Microcontroller Blockset. All the existing functionalities of the support package will be available in the new product.

SoC AXI master renamed as SoC AXI manager
Warns

SoC AXI master has been renamed to SoC AXI manager. In the software and documentation, the terms *manager* and *subordinate* replace *master* and *slave*, respectively.

These changes require updates to your code.

Type	Old Name	New Name	Recommendation
Objects	`socAXIMaster`	`socAXIManager`	`socAXIMaster` will be removed in a future release. Use `socAXIManager` instead. Replace all instances of `socAXIMaster` with `socAXIManager`.
AXI manager IPs	MATLAB as AXI Master	AXI Manager	The AXI Manager IP is upgraded from version 1.1 to 2.0.

R2022b

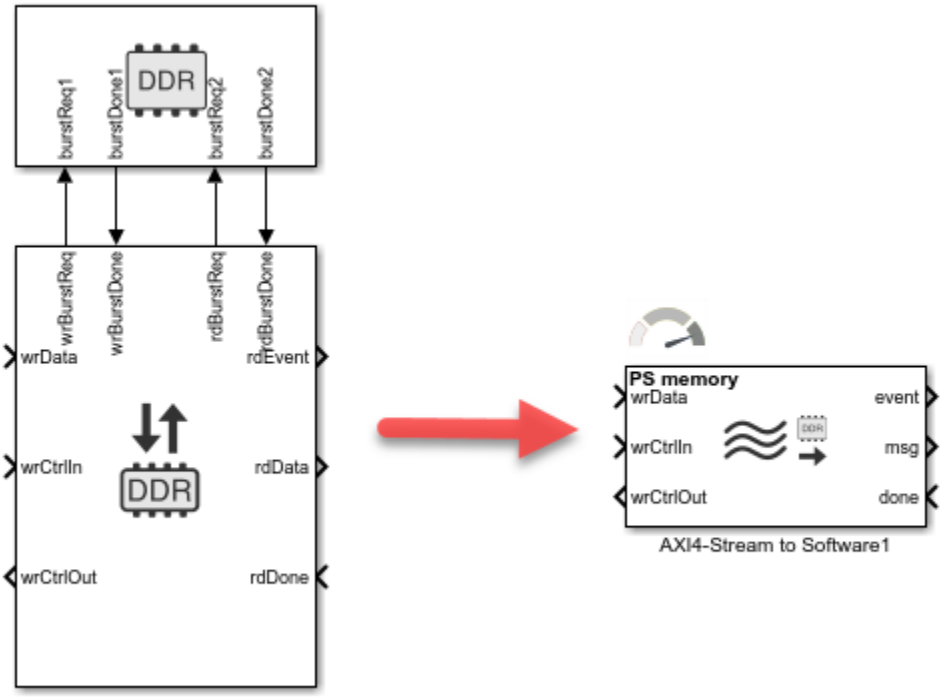
Version: 1.7

New Features

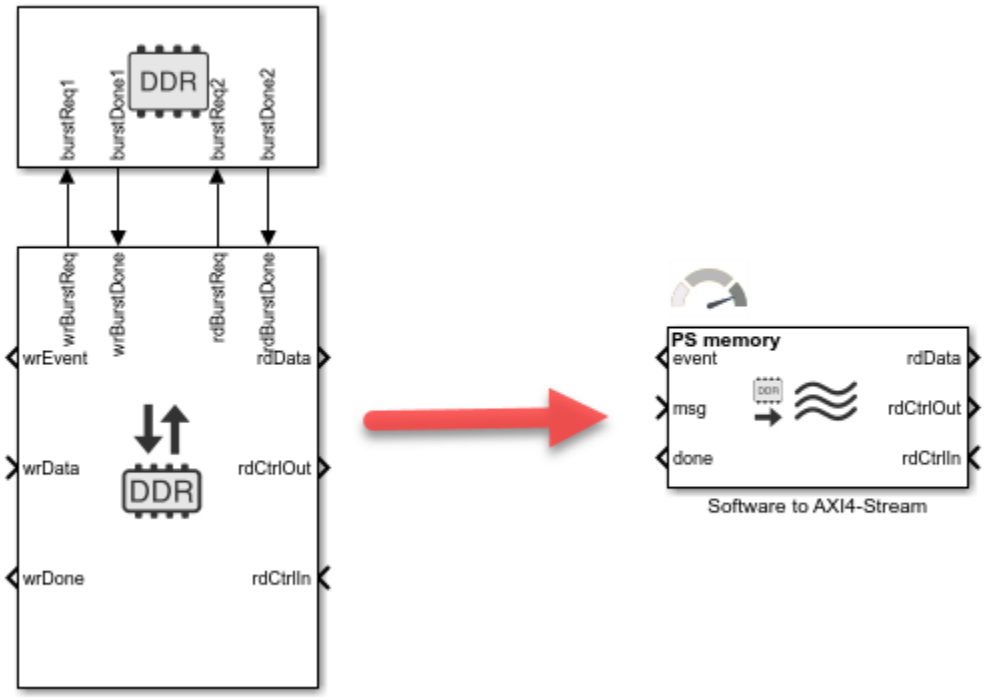
Memory library adds four blocks

SoC Blockset adds four blocks that replace a Memory Channel block connected to a Memory Controller block. These new blocks enhance simulation performance by streamlining your model with the introduction of a simple parameter set and fewer connections.

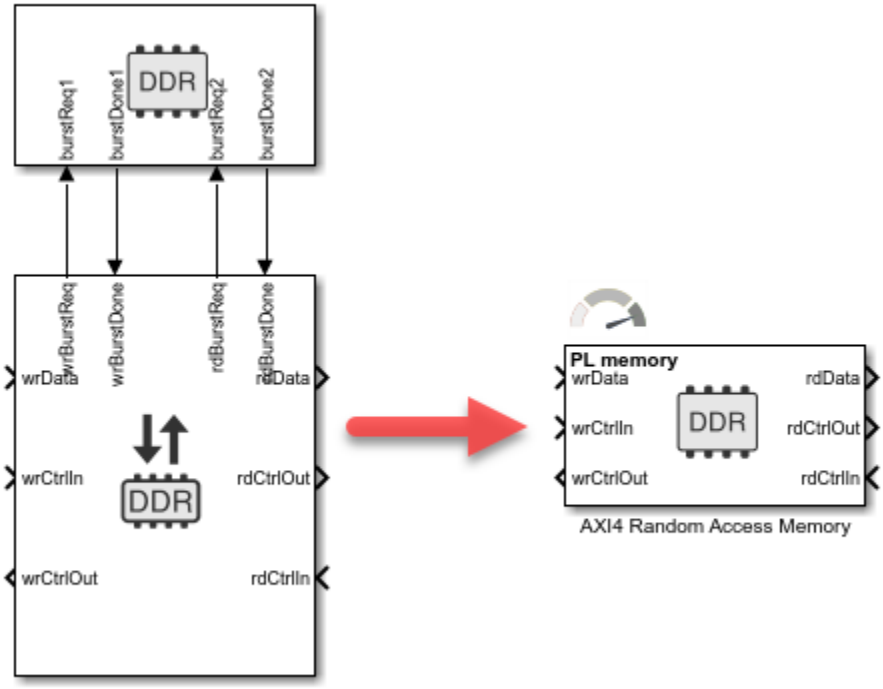
- AXI4-Stream to Software – Use this block to replace a Memory Channel block with the **Channel type** parameter set to `AXI4-Stream to Software via DMA` connected to a Memory Controller block.



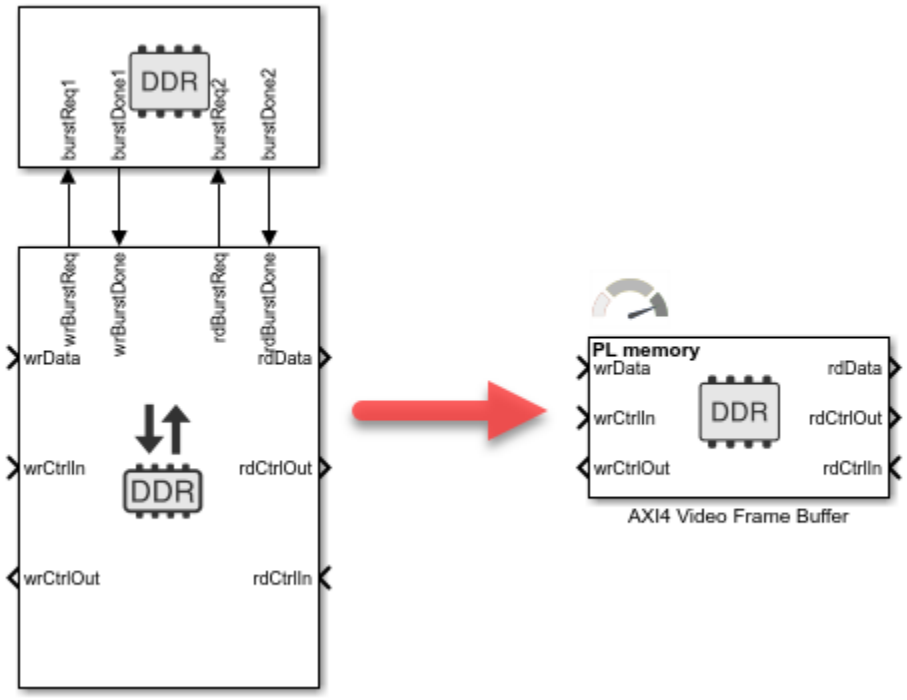
- Software to AXI4-Stream – Use this block to replace a Memory Channel block with the **Channel type** parameter set to `Software to AXI4-Stream via DMA` connected to a Memory Controller block.



- AXI4 Random Access Memory – Use this block to replace a Memory Channel block with the **Channel type** parameter set to `AXI4 Random Access` connected to a Memory Controller block.



- AXI4 Video Frame Buffer – Use this block to replace a Memory Channel block with the **Channel type** parameter set to `AXI4 Video Frame Buffer` connected to a Memory Controller block.



Memory Traffic Generator block adds option to hide ports

The Memory Traffic Generator block adds an option to show the input and the output ports connecting to the Memory Controller block. Clear this option when including one of the new simplified memory blocks in your design:

- AXI4 Random Access Memory
- Software to AXI4-Stream
- AXI4-Stream to Software
- AXI4 Video Frame Buffer

For more information, see the **`Show Memory Controller ports`** parameter in the Memory Traffic Generator block.

System Composer: System-on-Chip profiles, stereotypes, and scheduling analysis functions

SoC Blockset provides an SoC Blockset System Composer™ profile with the task and processor core stereotypes, as well as a scheduling analysis function. The stereotypes enable you to iterate quickly and make design decisions without the explicit functional system implementation or fully defined details.

The `soc_blockset_profile` is the `systemcomposer.profile.Profile` (System Composer) (System Composer) for an SoC Blockset system. This profile contains the `ProcessorCore`, `PeriodicSoftwareTask`, and `AperiodicSoftwareTask` stereotypes. Each stereotype contains a

set of properties, such as processor core count or task execution time, for the respective stereotype that maps to the property configurations available in the equivalent SoC Blockset model.

Schedule Editor: Support for event-driven tasks with lower priority than periodic rates

You can now specify event-driven tasks with lower priorities than the periodic tasks in the model. Task priority is now assigned based relative to the fundamental priority of the time-driven task. The relative priority assigned in code generation matches the relative priority defined in the **Schedule Editor** of the task partitions.

Hardware Mapping App

In R2022b, the new **Hardware Mapping** app combines the **Task Mapping** and **Peripheral Configuration** tools into a single app. The **Hardware Mapping** app unifies the process of setting hardware peripheral properties and assigning tasks to event or interrupt sources into one location during the application deployment.

You can assign task and peripheral properties manually using drop-down selections and edit boxes for the specific item. Also, tasks can be automatically assigned using an **Auto Map** button in the **Hardware Mapping** app toolstrip.

The **Hardware Mapping** app can be accessed from:

- New **Hardware Mapping** button on the SoC Blockset toolstrip
- A screen within the **SoC Builder** app
- Hardware Implementation Pane for SoC Blockset

R2022a

Version: 1.6

New Features

Compatibility Considerations

Task Manager Block Supports Task Execution Order Specification Using Scheduling Editor

Specify the execution order of tasks listed in the Task Manager block using the **Schedule Editor** app. The **Schedule Editor** app provides a visual representation of the execution order of the tasks, which map to reference model partitions.

Support of Texas Instruments Control Law Accelerator (CLA) Simulation

Simulate the behavior of a Control Law Accelerator (CLA) co-processor that allows parallel processing for timing sensitive tasks. Use the new CLA Task Manager block to simulate tasks in a CLA processor connected to a conventional C2000 processor managed by a Task Manager block.

Interprocess Data Channel Block Supports Propagation Delay and Buffer Modeling

The Interprocess Data Channel block now supports:

- Simulation of propagation delays of data transfers between two processors.
- Simulation of data buffer queue, including loss of data in the buffer queue.
- Diagnostics output port for visualizing buffer usage and data loss instances.
- Bus signal types.

Models containing the previous version of the Interprocess Data Channel block throws a warning on model update and provides option to update the block parameters to the new version.

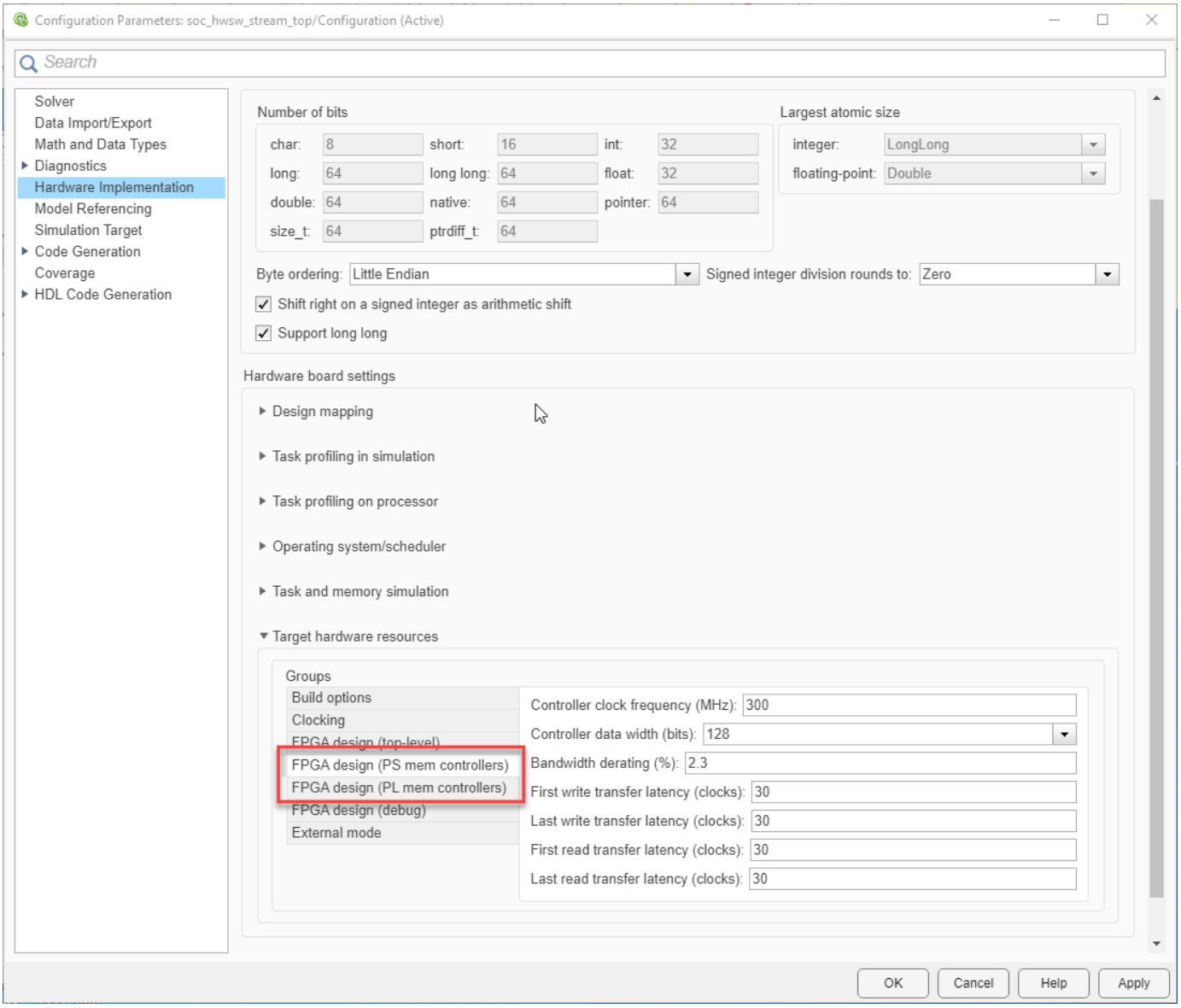
Support for Multiple Memory Controllers

You can now model a system with two memory units in the same architecture. This feature is helpful when you target Zynq boards such as Zynq 7000, Zynq SoC, or Zynq RFSoC, which support separate memory units for programmable logic (PL) and processing subsystem (PS).

When creating a model, you can instantiate up to two Memory Controller blocks. In the block mask, you can set **Memory selection** to `PS memory` or `PL memory`.

For an example that uses multiple memory controllers, see Receive Signal Waveform Using DDR4 on Xilinx RFSoC Device.

For supported boards, configure memory controller parameters for each board in the configuration parameters. Select **Hardware Implementation** in the left pane, and under **Hardware board settings** expand **Target hardware resource**. You can then set parameters separately for the PS memory and the PL memory. For more information about configuring memory controllers, see FPGA design (PS mem controllers).



Memory Channel Simulation Time Improvements

When simulating a model with a Memory Channel block, you can now improve simulation speed when a high-fidelity simulation is not required. In the block mask, select **Enable memory simulation** to simulate an accurate memory data path, or clear that parameter for faster simulation speed when debugging an algorithm. For an example that uses this feature, see Accelerate SoC Simulation by Varying Abstraction Levels.

Memory Channel Block Supports Sample Time Offset

You can now specify a sample time offset in the **Sample time** parameter of the Memory Channel block. Use the format `[period,offset]`.

- `period` – Time interval at which the block updates during simulation.
- `offset` – Time interval indicating an update delay. The block updates later in the sample interval than other blocks operating at the same period.

Previously, you could specify only a period and not an offset.

The Memory Channel block mask also has a new **Inherit sample time offset** parameter in the **Signal Attributes** tab. Select this parameter when the read data has the same offset as the write data.

Algorithm Analyzer: Support for variable-sized arrays

When you use `socFunctionAnalyzer` or `socModelAnalyzer` with a variable-sized array, the generated report now shows the runtime value for the array dimensions. If the runtime value changes during simulation, the report shows a range of values for the array size.

Updates to supported software

The SoC Blockset product now supports these software versions. For a full list of supported software, see Supported Third-Party Tools and Hardware.

- Intel Quartus Prime Standard 20.1.1
- Xilinx Vivado 2020.2

Updates to supported hardware

The SoC Blockset product now supports the Xilinx Zynq UltraScale+ RFSoC ZCU208 Evaluation Kit. For a full list of supported devices, see Supported Third-Party Tools and Hardware.

R2021b

Version: 1.5

New Features

Compatibility Considerations

Algorithm Analyzer functions support static analysis

The `socFunctionAnalyzer` and `socModelAnalyzer` functions now support static analysis for resources of MATLAB functions and Simulink models, respectively. Use the added `AnalysisMethod` name-value argument in either function to create a static-analysis report of resources.

Updates to ADC Interface block

The ADC Interface block now supports these features:

- Multichannel conversion — The ADC Interface block now supports multichannel conversions of the analog input signal. You can configure channels to sample sequential, simultaneous, or oversampling depending on the type and quality of the required measurement.
- Software generated ADC events - The block now supports software triggering of ADC events at the specified sample time rate.
- Interrupt-based event generation — The ADC Interface block can now conditionally enable an event output port that generates an interrupt event signal following the acquisition time or the acquisition time with conversion time of the analog to digital conversion.
- Analog watchdog event generation — The ADC Interface block now supports an event output port that generates a watchdog event signal when the analog input exceeds the specified **Lower threshold** or **Upper threshold** parameter boundary values.

Updates to PWM Interface block

The PWM Interface block now supports the following features:

- Event type generation — The PWM Interface block can now generate events on the ADC start or PWM interrupt.
- Event replication — The PWM Interface block can generate replica event ports, enabling other tasks or peripherals to be driven from a single PWM Interface block.

Digital I/O Interface block simulates hardware GPIO reads and writes of signals

The blockset now supports the simulation of general purpose input or output (GPIO) pins. The Digital I/O Interface block with the IO Data Sink or IO Data Source blocks can simulate the behavior of GPIO pins on the SoC or MCU.

Code Execution Profiling Report

Use the task analysis tool to generate an execution report from the latest simulation or external mode run of an SoC Blockset model. The execution report contains information on these:

- Task timing — Information includes the minimum, maximum, and average execution times of the tasks, total number of task executions over the duration of the simulation, and number of task overruns and task drops.
- CPU utilization — Information includes utilization of CPU by individuals tasks and total utilization of the available CPU cores.

SoC Model Creator and SoC Builder Tools: Create and deploy SoC model on RFSoC device

This workflow enables algorithm and system designers to generate an HDL IP core and integrate it into a fixed reference design for rapid prototyping. Create an SoC model based on the selected reference design by using the **SoC Model Creator** tool. Use the created model as a template to design and simulate an FPGA algorithm and processor algorithm. Then, implement and deploy the design on a Xilinx RFSoC device by using the **SoC Builder** tool. Xilinx Zynq UltraScale+ RFSoC ZCU111 and Xilinx Zynq UltraScale+ RFSoC ZCU216 evaluation kits support this workflow. For more information on this workflow, see RFSoC Support for Fixed Reference Design.

To use this feature, you must install the SoC Blockset Support Package for Xilinx Devices.

Support for C2000 F2838xD Processor

Simulate SoC models based on the Texas Instruments C2000 `F2838xD` family of processors. When combined with the SoC Blockset Support Package for Texas Instruments C2000 Processors you can:

- Generate embedded `C` code for SoC models, including code for device specific drivers.
- Evaluate portions of your design on hardware using Processor-in-the-Loop (PIL) simulation.
- Profile Real time on-device profiling.

To use the `F2838xD`, select the `TI F2838xD (SoC)` option for **Hardware board** in the **Configuration Parameters** window. For more information on the TMS320F28388D processor, see https://www.ti.com/product/TMS320F28388D

Functionality being removed or changed

SoC Model Creator tool replaces removed SoC Template Builder tool
Behavior change

The **SoC Template Builder** tool has been removed. Instead, use the **SoC Model Creator** tool to create an SoC model based on the selected reference design. The **SoC Model Creator** tool has these advantages over the removed **SoC Template Builder** tool.

- Create a template model with the interfaces that are compatible with the reference design.
- Create these types of template models.
  - FPGA and processor
  - FPGA and memory
  - FPGA only

To open the **SoC Model Creator** tool, execute the `socModelCreator` command in the MATLAB command prompt. Using the `socTemplateBuilder` command, which previously opened the **SoC Template Builder** tool, is not recommended. If you use the `socTemplateBuilder` command in the MATLAB command prompt, the command now opens the **SoC Model Creator** tool.

To use the **SoC Model Creator** tool, you must install the SoC Blockset Support Package for Xilinx Devices.

R2021a

Version: 1.4

New Features

Bug Fixes

Memory Channel block supports vectors up to 512 bits

The Memory Channel block now supports vectors up to 512 bits. This feature enables you to create:

- A multichannel frame
- A multichannel sample
- A video frame with multiple pixels
- A video sample with multiple pixels
- A vectorized frame
- A vectorized pixel

To enable sample packing, open the Memory Channel block mask, and on the **Signal Attributes** tab, select the **Enable sample packing (last signal dimension as channel)** parameter.

When you select this parameter, the Memory Channel block packs data for the last dimension of the signal. For example, if the channel data-type is `uint32`, the dimensions are `[1024,4]`, and you select **Enable sample packing (last signal dimension as channel)**, then the memory channel generates 1024 read or write transactions of 128 bits. For this example, if you clear this sample packing parameter, the memory channel generates 4096 transactions of 32 bits each.

The combined width of the flattened signal must not exceed 512 bits. For more information, see Memory Channel.

Memory Channel block simulates pipelined read and write operations

The Memory Channel block now simulates pipelined random access read and write operations when you set the **Channel type** to `AXI4 Random Access`. This feature enables a higher simulation fidelity of the memory random access throughput. For more information about the AXI4 interface see Simplified AXI4 Master Interface.

Memory Channel block supports extended data types

The Memory Channel block now supports 16-bit and 128-bit integer data types when you set the **Channel type** parameter to `Software to AXI4 Stream via DMA` or `AXI4 Stream to Software via DMA`.

Audio blocks simulate capture and playback of hardware audio

The blockset now supports the simulation of hardware audio capture and playback. The Audio Playback Interface and Audio Playback blocks simulate the output of audio signals to one of the specified audio hardware peripherals in supported SoC devices. Similarly, the Audio Capture Interface and Audio Capture blocks simulate the capture of audio signals from one of the specified audio hardware peripherals in supported SoC devices. When you use the Audio Playback and Audio Capture blocks with the SoC Blockset Support Package for Embedded Linux® Devices, the blocks generate driver code that works with Linux audio devices that comply with **Advanced Linux Sound Architecture** (**ALSA**) .

Video blocks simulate capture and display of hardware video

The blockset now supports the simulation of hardware video capture and display. The Video Display Interface and Video Display blocks simulate the output of video signals to one of the specified video

outputs in supported SoC devices. Similarly, the Video Capture Interface and Video Capture blocks simulate the capture of video signals from one of the specified video peripherals in supported SoC devices. When you use the Video Display and Video Capture blocks with the SoC Blockset Support Package for Embedded Linux Devices, the blocks generate driver code that works with Linux video devices that comply with **Video4Linux (V4L2)** and **Simple Directmedia Layer (SDL)**.

PWM blocks support multiple comparators, variable frequency and variable phase inputs

The PWM Write and PWM Interface blocks now support up to two independent comparators for refined PWM output generation along with events. Comparator-generated events, when connected to the Task Manager block, provide more fine grained control over one PWM output cycle. The PWM blocks now supports explicit phase and frequency input to provide offset synchronization between multiple PWM modules and changing PWM frequency at runtime.

Updates to supported software

SoC Blockset now supports Xilinx Vivado 2020.1. For a full list of supported software, see Supported Third-Party Tools and Hardware .

Updates to supported hardware

SoC Blockset now supports the Zynq UltraScale+ RFSoC ZCU216 Evaluation Kit. For a full list of supported devices, see Supported Third-Party Tools and Hardware .

R2020b

Version: 1.3

New Features

Bug Fixes

AXI Stream: Stream data from software to a hardware IP

SoC Blockset now supports a memory protocol to stream data from software to an FPGA. Use the new Stream Write block to initiate and drive a stream from the software subsystem to the FPGA algorithm. To configure the memory channel for this mode, in the Memory Channel block set the **Channel type** parameter to `Software to AXI4-Stream via DMA`.

FPGA IP Core interrupt: Trigger an event-driven software task via interrupt

You can now issue an interrupt from an FPGA or a peripheral IP to software. Use the Interrupt Channel block to arbitrate between interrupt requests, process them one at a time, and send the requests to the Task Manager block for software processing.

ADC Interface Block: Simulate hardware ADC of signals

The ADC Interface and ADC Read blocks simulate the analog-to-digital conversion (ADC) of signals of hardware peripherals in supported SoC devices.

PWM Interface Block: Simulate hardware PWM output signals

The PWM Interface and PWM Write blocks simulate the pulse-width-modulation (PWM) output signals of hardware peripherals in supported SoC devices.

Multiprocessor Simulation: Simulate SoC models with multiple processors

The SoC Blockset software now supports the simulation of multiple processors contained in a single SoC. To simulate a specific processor in the SoC, you can uniquely configure the hardware settings of each Reference block mode. When you use the reference models in combination with a hardware support package, you can deploy the reference models to the specific processors on the SoC hardware.

Interprocessor Communication (IPC): Simulate communication between multiple processors

You can simulate IPC by using the Interprocess Data Read, Interprocess Data Channel, and Interprocess Data Write blocks. The Interprocess Data Channel block simulates hardware behavior for messages sent from one reference processor and received by another processor. The Interprocess Data Write block sends the messages, and the Interprocess Data Read block reads the received messages.

Task Manager Block Enhancements: Deploy tasks as ISRs on bare metal devices

The Task Manager block now supports the simulation and code generation of interrupt service routines (ISR) for bare metal devices. To simulate and manage tasks triggered by hardware interrupts

on systems without an operating system (OS), use the Task Manager block. Interrupts can be produced by the ADC Interface block and from FPGA through the ISR block.

Model Templates: Use added templates or enhanced existing template to design required SoC model

These added model templates are now available.

- RFSoC Template – Use this template to design an RFSoC-enabled wireless communication SoC model.
- Stream from Processor to FPGA Template – Use this template to design a data path from software (Processor) to hardware (FPGA).

Also, the existing Frame Buffer with HDMI Template model template has been enhanced. For more information on model templates, see Use Template to Create SoC Model.

Visualize logged Stateflow states in the Logic Analyzer

When you log signals in Stateflow® charts, you can use the **Logic Analyzer** to visualize the state changes. To log Stateflow states, in the **Simulation** tab, under **Prepare**, select a state logging option. In the Logic Analyzer, you'll see your states marked for logging in the left column.

SoC Blockset Support Package for Texas Instruments C2000 Processors: Generate, build, and deploy reference designs on TI’s C2000 processors

The SoC Blockset Support Package for Texas Instruments C2000 Processors with Embedded Coder® can export reference designs to TI’s C2000™ devices. These reference designs can be used with Texas Instruments design tools.

This support package helps to automate the integration, execution, and verification of reference designs for the hardware boards using TI’s C2000 processors. Supported hardware boards include the Texas Instruments C2000Delfino MCU F28379D LaunchPad and Texas Instruments C2000 Delfino MCU F2837xD Control Card. For more information on these hardware boards, see https://www.ti.com/tool/LAUNCHXL-F28379D and https://www.ti.com/tool/TMDSCNCD28379D, respectively.

SoC Blockset Support Package for Embedded Linux Devices: Design, analyze, and prototype for embedded Linux devices

SoC Blockset Support Package for Embedded Linux Devices enables you to model, simulate, analyze, and prototype software on embedded Linux platforms using SoC Blockset. The support package features key capabilities including embedded C code generation with POSIX® threads and rate-monotonic scheduling (RMS), device driver integration, and device tree and Linux image customization.

Use IP core generation workflow for RFSoC devices (November 2020, Version 20.2.1)

Generate an RFSoC model template using **SoC Template Builder**, modify it to create an RFSoC IP, and then use IP core generation workflow to deploy the IP on a Xilinx RFSoC device. This workflow is

supported for Zynq UltraScale+ RFSoC ZCU111 boards. For additional information, see “Configure Design Using SoC Model Creator” (SoC Blockset Support Package for Xilinx Devices).

Updates to supported software

SoC Blockset now supports Xilinx Vivado 2019.2. For a full list of supported software, see https://www.mathworks.com/help/releases/R2020b/soc/ug/supported-third-party-tools-and-hardware.html.

R2020a

Version: 1.2

New Features

Bug Fixes

Compatibility Considerations

socExportReferenceDesign Function: Automate the creation of custom reference design

Use the `socExportReferenceDesign` function to export a custom reference design from your SoC Blockset Simulink model. You can then use IP Core Generation workflow (requires HDL Coder™) to generate a custom IP core and integrate it into your SoC reference design. For more information about the `socExportReferenceDesign` function, see `socExportReferenceDesign`.

SoC Algorithm Analysis Functions: Analyze MATLAB functions or Simulink models early in the design process

Use the `socFunctionAnalyzer` and `socModelAnalyzer` functions to analyze and compare different algorithms and models early in the design process. The `socFunctionAnalyzer` and `socModelAnalyzer` functions create a report detailing the number of operations in a MATLAB function or Simulink model, respectively.

IP Core Register Read and Register Write Blocks: Model write operation from processor to hardware logic

The Register Write block and IP Core Register Read block model a write operation from a processor to hardware logic. The IP Core Register Read block receives data sent by a Register Write block from the processor. You can replace the Register Channel block with the IP Core Register Read block when writing from the processor to the FPGA, eliminating the need to wire signals through the model hierarchy. For an example using IP Core Register Read and Register Write blocks, see Packet-Based ADS-B Transceiver.

Memory Channel Block Enhancements: Extended data-type support

The Memory Channel block now supports 64-bit integer data types and up to 128-bit fixed-point data types.

I2C Master Block Enhancements: Extend configuration support to additional slave devices and support for clock stretching feature

The I2C Master block now provides configuration support to I2C slave devices with a 10-bit address. Using the clock stretching feature, the block can now synchronize with slow-running I2C slave devices. The block also supports a burst mode with a maximum of 256 bytes of data per transaction and provides a **statusReg** port to read the status of an I2C bus.

For compatibility considerations, see “I2C Master block has one data register input port and one data register output port” on page 7-3.

socBuilder Tool Enhancements: Improve validation checks and status messages

The **socBuilder** tool generates an FPGA validation report in HTML format. This report lists the warnings and errors, if any, encountered during model parsing. The tool now displays improved

status messages on the **SoC Builder > Build Model** wizard while performing synthesis, implementation, and bit-stream generation processes.

IO Data Source Block: Read data from timeseries object in MATLAB workspace

The IO Data Source block now supports `timeseries` object in the MATLAB workspace as an input source. This feature enables you to provide custom data along with time of occurrence as an input to the block.

Proxy Task Block: Support event-driven tasks

The Proxy Task block now supports event driven task triggers from the Task Manager block. For more information of event driven tasks, see Event-Driven Tasks.

Kernel Profiler: Perform kernel instrumentation profiling for unlimited duration

Previously, the kernel profiler performed kernel instrumentation profiling on the hardware for only a limited duration (based on the free disk storage available on the hardware). Now, the kernel profiler provides an option to perform kernel instrumentation profiling for unlimited time duration.

Updates to supported software

SoC Blockset now supports Xilinx Vivado 2019.1. For a full list of supported software, see Supported Third-Party Tools and Hardware.

Updates to supported hardware

SoC Blockset now supports Zynq UltraScale+ RFSoC ZCU111 Evaluation Kit. To simulate and prototype radio frequency (RF) effects with the RF Data Converter block, use the SoC Blockset Support Package for Xilinx Devices features. For details about this support package, see SoC Blockset Support Package for Xilinx Devices. For a full list of supported devices, see Supported Third-Party Tools and Hardware.

Functionality being removed or changed

I2C Master block has one data register input port and one data register output port

Behavior change

In R2019a, the I2C Master block has input data register ports **dataReg**, **dataReg1**, **dataReg2**, and **dataReg3** and output response data register ports **respData**, **respData1**, **respData2**, and **respData3**. These data register ports support a maximum of 16 bytes per transaction. In R2020a, these register ports are replaced with single data register ports: input port **dataReg** and output port **respData**. Each of these single data register ports support a maximum of 256 bytes per transaction.

In R2020a, Simulink errors if you open a model that was created in an earlier release and that contains an I2C Master block. In this case, connections to ports **dataReg1**, **dataReg2**, **dataReg3**, **respData1**, **respData2**, and **respData3** are either missing or reconnected to empty ports on the

block. Manually check and update the port connections in your model to proceed further. For more information, see I2C Master block documentation.

Signal data no longer streams to the Logic Analyzer when signal logging is disabled
Behavior change

Previously, signals marked for logging have streamed to the **Logic Analyzer**, regardless of the setting for **Signal logging** in the model configuration parameters. Starting in R2020a, signals marked for logging stream to the Logic Analyzer *only* when signal logging is enabled for a model.

To view data in the Logic Analyzer, you must enable signal logging for the model. (Logging is on by default.) To enable signal logging, open **Model Settings** from the toolstrip, navigate to the **Data Import/Export** pane, and select **Signal logging**.

R2019b

Version: 1.1

New Features

Bug Fixes

Proxy Task Block: Model the effect of a task in your application without an explicit task implementation

The Proxy Task block models the effect of a task on your application without defining the implementation of the task. You can configure the block execution timing using the Task Manager block. The Proxy Task block serves as a placeholder for a task you are currently developing or plan to develop.

Testbench Task Block: Model the effect of your external task competing for resources with an application

The Testbench Task block models the effect of a task in an external application competing for execution resources with the tasks in your application. You can configure the block execution timing by using the Task Manager block.

Playback control behavior changed for Logic Analyzer in referenced models

When you use the **Logic Analyzer** in a referenced model, the playback controls in the Logic Analyzer now match the playback controls of the last model you interacted with that logs data to the scope. For example, if you opened the Logic Analyzer from a model referenced by another model with the Model block, the run button  in the scope runs the top level model. If the referenced model is opened as a top model, the run button runs the referenced model in isolation.

Updates to supported software

SoC Blockset now supports to these software versions. For a full list of supported software, see Supported Third-Party Tools and Hardware.

- Xilinx Vivado 2018.3
- Intel Quartus Prime 18.1

IO Data Source Block: Read data from a recorded data file at the same time interval at which it was recorded on the hardware board

You can now use the IO Data Source block to read data from a recorded data file at the same time interval as it was recorded on the hardware board.

When you enable the event port of the IO Data Source block and connect it to the Task Manager block, the IO Data Source block reads event signals and generates them on the event port as they were recorded in the recorded data file. Previously, the IO Data Source block generated events based on the sample time specified on the block mask.

This feature enables you to see the real behavior of the data from hardware I/O peripherals in Simulation.

Kernel Profiler: Monitor and record execution times of tasks with LTTng

You can use a Kernel profiler to monitor and record model-related processes and threads running on the Linux of your hardware board without instrumenting code.

Hardware Memory Diagnostics: View additional latencies and data overflow information from FPGA execution

In addition to viewing **Burst request to first transfer complete** latency information from a design running on field programmable gate array (FPGA), you can now view **Burst execution** and **Burst last transfer to complete** latencies information. This feature is similar to that of Simulation Performance Plots.

You can also collect and view data overflow that occurred in bandwidth, burst, and latency memory diagnostics metrics.

R2019a

Version: 1.0

New Features

Introducing SoC Blockset: Design, evaluate, and implement SoC hardware and software architectures

SoC Blockset provides Simulink blocks and visualization tools for modeling, simulating, and analyzing hardware and software architectures for ASICs, FPGAs, and systems on a chip (SoC). You can build your system architecture using memory models, bus models, and I/O models, and simulate the architecture together with the algorithms.

SoC Blockset lets you simulate memory and internal and external connectivity, as well as scheduling and OS effects, using generated test traffic or real I/O data. You can quickly explore different system architectures, estimate interface complexity for hardware and software partitioning, and evaluate software performance and hardware utilization.

SoC Blockset exports reference designs for Xilinx and Intel FPGA devices and SoC platforms, including Zynq-7000, UltraScale+, and Intel SoC FPGAs. These reference designs can be used with Xilinx and Intel design tools.

SoC Blockset Support Package for Xilinx Devices: Generate, build, and deploy reference designs on Xilinx devices

The SoC Blockset Support Package for Xilinx Devices with Embedded Coder or HDL Coder can export reference designs for Xilinx FPGA devices and SoC platforms. These reference designs can be used with Xilinx design tools.

The support package helps to automate integration, execution, and verification of reference designs for the SoC platforms, including Xilinx Artix®-7, Xilinx Kintex®-7, XilinxZynq, and XilinxZynqUltraScale+.

SoC Blockset Support Package for Intel Devices: Generate, build, and deploy reference designs on Intel devices

The SoC Blockset Support Package for Intel Devices with Embedded Coder or HDL Coder can export reference designs for Intel FPGA devices and SoC platforms. These reference designs can be used with Intel design tools.

The support package helps to automate integration, execution, and verification of reference designs for the SoC platforms, including Intel Arria® 10 and Intel Cyclone® V.