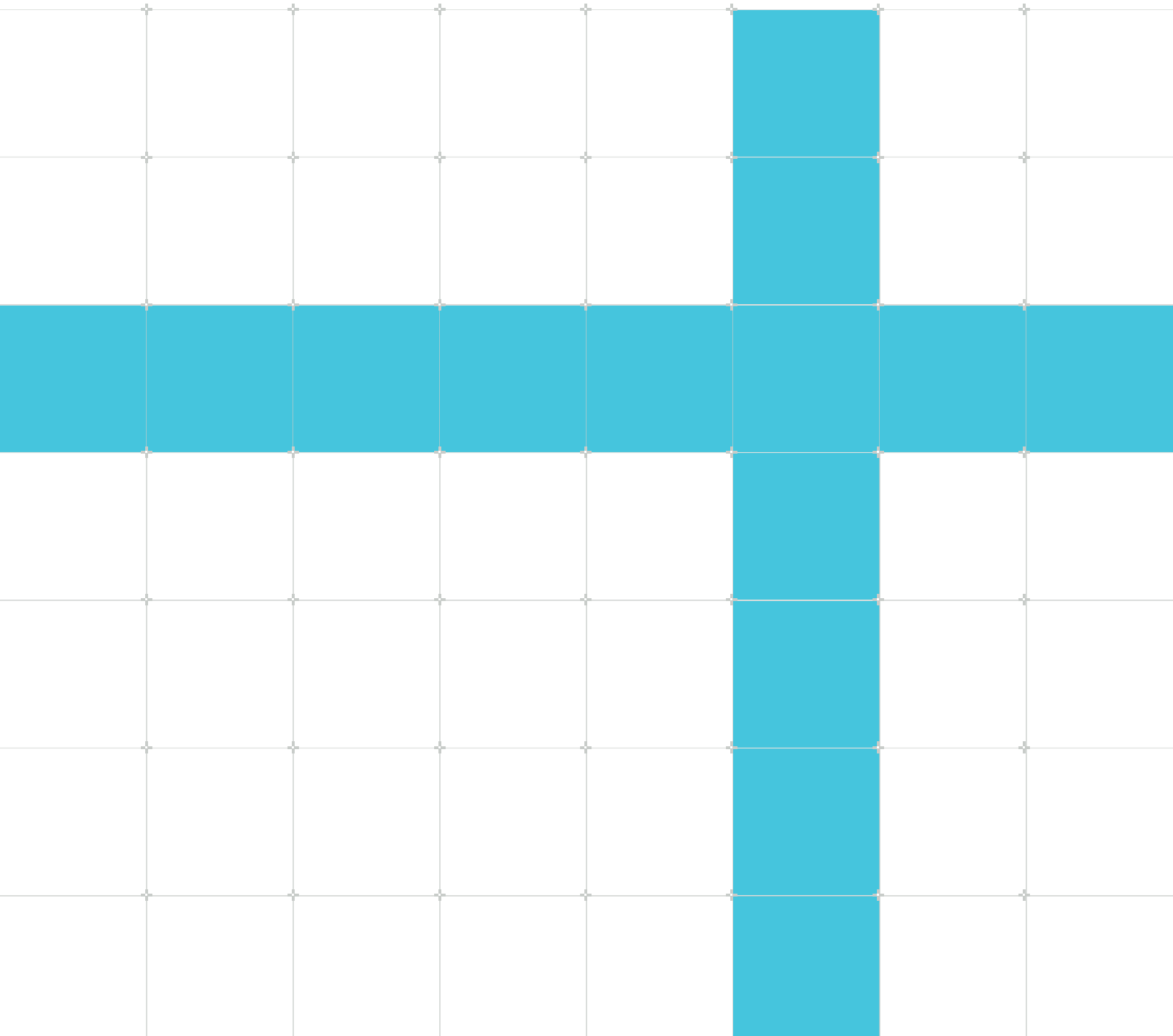
Fixed Virtual Platforms for Arm® Corstone™ SSE-320

Revision r0p0

Reference Guide

Non-Confidential

Issue 01
109760_0000_01_en



Fixed Virtual Platforms for Arm® Corstone™ SSE-320 Reference Guide

This document (109760_0000_01_en) was issued on 2024-09-27. There might be a later issue at http://developer.arm.com/documentation/109760

The product revision is r0p0.

See also: Proprietary notice | Product and document information | Useful resources

Start reading

If you prefer, you can skip to the start of the content.

Intended audience

This book is written for electronics engineers that are designing, configuring, laying out, validating, and verifying the Arm® SSE-320 Example Subsystem. It is assumed that the primary readers have high technical ability and have experience of Verilog. The documentation does not assume experience of Arm devices or implementation, but it is likely that the primary readers have some prior experience of Arm IP. It is assumed that they are familiar with all necessary implementation tools and compute resources. It is also assumed that the readers have access to all necessary tools.

Inclusive language commitment

Arm values inclusive communities. Arm recognizes that we and our industry have used language that can be offensive. Arm strives to lead the industry and create change.

We believe that this document contains no offensive language. To report offensive language in this document, email terms@arm.com.

Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on the product, create a ticket on https://support.developer.arm.com.

To provide feedback on the document, fill the following survey: https://developer.arm.com/documentation-feedback-survey.

Contents

1. Overview of FVPs

Fixed Virtual Platforms (FVPs) enable software development without the need for real hardware. They are software models that closely imitate a modelled component or subsytem. They also model the FPGA test boards that are available for these products. Any deviations from the modelled product behavior are documented. Most are available for both Linux and Windows hosts.

Depending on your requirements, you can:

- Use source code examples that are provided in the Fast Models package, with the tools required to customize and build them. These are not documented in this guide. For more information, see the *Fast Models Reference Guide*.
- Use a FVP that is supplied as pre-built executables. Their composition is fixed, but you can configure their behavior using parameters.

Most Fast Models FVPs are provided in a single license-managed library. Some Architecture Envelope Model (AEM) FVPs are available for download without requiring a license.

FVPs are available for all Cortex®-A, Cortex®-R, Cortex®-M, and Neoverse™ processors, and most of them support the MTI, and Iris interfaces, so they can be used for debugging and for trace output.

The scope of this document is the pre-built Arm® Corstone™ SSE-320 Subsystem FVP, which is downloadable from Arm Developer. No license is required.

Arm provides validated Linux and Android deliverables for FVPs.

- For details, see the Arm Development Platforms wiki on Arm Community.
- To get started with Linux on FVPs, see FVPs on Arm Community.

Related information

- Contents of the FVP Standard Library package

1.1 FVP requirements

Most FVPs can run on either Windows or Linux host machines.

Host machine

Architecture

x86-64 and Arm AArch64 host platforms are supported.

Minimum specification

At least 2GB RAM, preferably 4GB.

2GHz Intel Core2Duo, or similar, that supports the MMX, SSE, SSE2, SSE3, and SSSE3 instruction sets.

Recommended specification

At least double the RAM of the platform you intend to simulate. For example, a simulated platform containing 8GB of DRAM should be run on a 16GB host machine.

Fast Models and associated FVPs benefit most from high single-threaded performance. For example, a high frequency (4-5GHz) Intel Core i9 or i7 or AMD Ryzen 9 or 7 host CPU gives a significant improvement, between 30-60%, over Intel Xeon cores (2-3 GHz).

Operating system

Linux

Red Hat Enterprise Linux 7 or 8 (for 64-bit architectures), Ubuntu 18.04, 20.04, or 22.04 Long Term Support (LTS).

Windows

Microsoft Windows 10 64-bit.

Compiler

FVPs are built with Visual Studio 2019 and GCC 9.3.0.

1.2 Contents of the FVP Standard Library package

The FVP Standard Library consolidates commonly used FVPs into a single package which also contains some useful plug-ins and utilities.

- The FVPs are supplied as pre-built executables. For the source code for these and other FVPs, install the main Fast Models product.
- The package does not include unlicensed FVPs. These are available for download separately, from Arm Architecture Models on Arm Developer.

The package installs the FVPs under the models directory. It also installs:

Plug-ins

Plug-ins are DLLs or shared objects that provide extra functionality to FVPs, for instance different types of trace output. To load a plug-in, pass the name of the plug-in to the FVP at startup using the `--plugin` command-line option or using the FM_TRACE_PLUGINS environment variable.

For more information about plug-ins, see the Plug-ins for Fast Models section in the *Fast Models Reference Guide*.

Model Shell and Model Debugger

Model Shell is a command-line tool for launching FVPs. For more information about Model Shell, see Model Shell for Fast Models Reference Guide. Model Debugger is an easy to use

symbolic debugger with a GUI that allows you to debug software running on the FVP. For more information about Model Debugger, see Model Debugger for Fast Models User Guide. They are installed in the `bin` directory.

iris.debug Python module

`iris.debug` is a Python scripting interface to Fast Models. It allows you to interact with FVPs, including connecting to and configuring them, performing execution control, and accessing registers and memory. It is installed under the Iris directory. For more information about iris.debug, see Iris Python Debug Scripting User Guide.

2. Getting started with Fixed Virtual Platforms

There are different ways to launch an FVP, for example from the command prompt, or from Model Debugger or Arm Development Studio.

Downloading an FVP

The SSE-320 FVP can be downloaded from Arm Ecosystem FVPs on Arm Developer.

Running an FVP from command prompt

To run an FVP from the command prompt, enter the model name followed by:

- One or more -C command-line arguments. To see all available options, use the –help option.
- A configuration file and the -f command-line argument.

Each -C command-line argument or line in the configuration file must contain:

- The name of the component instance.
- The parameter to modify.
- Its value.

Use the following format: `instance.parameter=value`

The instance can be a hierarchical path, with each level separated by a dot . character.

- Comment lines in the configuration file begin with a # character.
- You can set Boolean values using either true or false, or 1 or 0.

You can generate a configuration file with all parameters set to default values by using the `-o` option to redirect the output from the `--list-params` option, for example: `FVP_Base_AEMvA.exe --list-params -o params.txt`

Python environment for VSI/VIO/VSOCKET

FVPs that include the AVH peripherals, for example, the MPS4 platforms, require the standard Python libraries. These libraries are provided by the Fast Models FVP library installation.

To set up the environment variables for the FVPs, including automatically configuring the FVPs to use the standard Python libraries:

Run the command: `source $FVP_INSTALL_PATH/scripts/runtime.sh`

Alternatively, manually set the `PYTHONHOME` environment variable to the location of the standard Python libraries.

Related information

- FVP command line options

2.1 FVP command line options

Specify these options when you launch an FVP from the command line. You can specify these options in any order.

Iris-related options

`-iris-connect conspec`

Start the Iris server according to the connection specification conspec. conspec is a structured string argument that can contain flags and parameters.

Use `-iris-connect help` to print a list and description of all available Iris connection types.

The following options are ignored when using `-iris-connect`:

- `-iris-allow-remote`
- `-iris-port`
- `-iris-port-range`

To set these connection parameters, use:

```
-iris-connect tcpserver\[,port=PORT\]\[,endport=ENDPORT\]\[,allowRemote\]
```

This command starts the TCP server on the first free port in the range PORT-ENDPORT, where the default for PORT is 7100 and the default for ENDPORT is PORT + 63. Only local connections are allowed, unless allowRemote is specified.

The other supported connection type is socketfd=FD which uses the socket file descriptor FD as an established UNIX domain socket connection.

Use `-iris-connect verbose=n` to set the logging level of the IrisTcpServer, wheren is 0-3.

`-iris-log (-i)`

Log to stdout all Iris function calls that were made during the simulation.

There are 5 possible log levels. To set a level greater than 1, specify the option multiple times, for example -ii for level 2.

The log levels have the following meanings:

0

Logging is disabled. This value is the default.

1

Log messages use a compact single-line format.

2
: Log messages use a single-line pseudo-JSON format.

3
: Log messages use a more readable, multi-line pseudo-JSON format.

4
: As 3 but also prints the U64JSON hex value of the message.

To set the Iris log level for all FVP invocations, use the IRIS_GLOBAL_INSTANCE_LOG_MESSAGES environment variable.

`-print-port-number (p)`
: Print the port number on which the Iris server is listening.

Use this option with `-iris-server`

This option can be useful if you need to specify the port number when you connect a client to the debug server.

Output-related options

`-list-instances`
: Print a list of model instances to standard output, then exit the simulation.

  Use this option to help identify the correct syntax for configuration files, and to find out what instances the target supplies.

`-list-params (-l)`
: Print a list of the available model parameters and their default values to standard output, then exit the simulation.

If you are loading a plug-in, this option also lists the plug-in parameters.

`-list-set-params <filename>`
: Save a list of the set model parameter values for each component to a file. If filename is -, print to standard output instead.

  An alternative to this command-line option is to use the FASTSIM_LOGPARAMS and FASTSIM_LOGPARAMS_FILE environment variables:

- To log parameter values to stdout, set FASTSIM_LOGPARAMS to 1.
- To log parameter values to a file, additionally set FASTSIM_LOGPARAMS_FILE to the name of the file.

-dump-params

Dump the list of model parameters into a JSON file called parameter_list.json, then exit the simulation. The file is created in the current working directory.

-list-regs

Print model register information to standard output, then exit the simulation.

-output <filename> (-o)

Redirect output from the `-list-instances`, `-list-memory`, `-list-params`, and `-listregs` commands to a file. If this option is used with `-list-params`, the contents of the output file are formatted correctly for use as input by the `-config-file` option.

-log filename

Log all SystemC reports into filename.

-stat

Print the following performance statistics on simulation exit:

Simulated time

An estimate of the time that the workload would have taken on the modeled hardware.

User time

Time in wall clock seconds that the host CPU spent running in user mode.

System time

Time in wall clock seconds that the host CPU spent running in system mode.

Wall time

Time in wall clock seconds between the simulation starting and stopping.

Performance index

An estimate of the accuracy of the simulation performance. This value is Simulated time divided by Wall time.

-prefix (-p)

Prefix each line of semihosting output with the name of the target instance.

-help (-h)

Print the help message and exit.

-version

Print version information for the FVP.

-quiet (-q)

Suppress informational output.

-keep-console (-K)

Keep the console window open after completion. This option applies to Windows hosts only.

-disable-model-exitcode

Disable the simulation from retrieving the exit code returned by a model or a plug-in. By default, it is enabled.

Run control options

-cpulimit n

Maximum number of wall clock seconds for the simulation process to be active. This value excludes simulation startup and shutdown.

This option is ignored if a debug server is started.

The default is unlimited.

-cyclelimit n

Maximum number of cycles to run.

This option is ignored if a debug server is started.

The default is unlimited.

-timelimit n (-T)

Maximum number of wall-clock seconds for the simulation to run, excluding startup and shutdown.

To terminate the model immediately after initialization, specify `-timelimit 0`.

-simlimit n

Maximum number of seconds to simulate.

This option is ignored if a debug server is started. The default is unlimited.

Like the Simulated time value output by `-stat`, this value is measured in simulation seconds, not wall-clock seconds.

-break [instance=][threadid:][memspace@]address (-b)

Set a program breakpoint on the address of an instruction, where:

- threadid is an optional thread id, for a thread-specific breakpoint.
- memspace is an optional name or id of the memory space that address is in. If not specified, the breakpoint is set on the first program memory space found.
- If the FVP has multiple cores, you must specify an instance, for example: `\-b FVP_Base_AEMvA.cluster0.cpu0=0x800010eC`

This option can be specified multiple times.

Timing and performance options

-cpi--file filename

Use filename to set the Cycles Per Instruction (CPI) class.

-quantum n (-Q)

Number of ticks to simulate for each quantum. The default is 10000.

-min-synclatency n (-M)

Number of ticks to simulate before synchronizing. Events that occur at a higher frequency than this value are missed. The default is 100.

-fast-ram filename

Enable FastRAM and load the configuration from filename.

Memory-related options

-dumpfile @address,size

Dump a section of memory to a file at model shutdown. This option can be specified multiple times. The full syntax is: `-dump [instance=]file@[memspace:]address,size`

To see the list of instances and memory spaces, use the `--list-memory` option.

-data file@address

Print model memory information to standard output, then exit the simulation.

-start [instance=]address

Set the initial PC value to this address, overriding the .axf start address.

- Use this option if you do not want the CPU to start executing at the default reset address.

  You do not normally need to do this if you are loading an ELF file using `-application`.
- This option can be used with `--data` to load binary data that is not in an ELF file.

Configuration options

-parameter instance.parameter=value (-C)

Set a parameter. This option can be specified multiple times. Specify the full hierarchical name of the parameter.

This option is also used to set plug-in parameters.

-config-file filename (-f)

Load the parameters from a configuration file.

Options for loading a plug-in or application

`-parameter instance.parameter=value (-a)`

Load an application. On a multi-core system, specify the instance, or use * to load the application image into all the cores in a cluster:

```
-a cluster0.cpu*=file
```

`-plugin filename`

Load the plug-in filename. This option can be specified multiple times. You can also load plug-ins using the FM_TRACE_PLUGINS environment variable.

For information about plug-ins, see the Plug-ins for Fast Models section in *Fast Models Reference Guide*.

2.2 Using the CLCD window

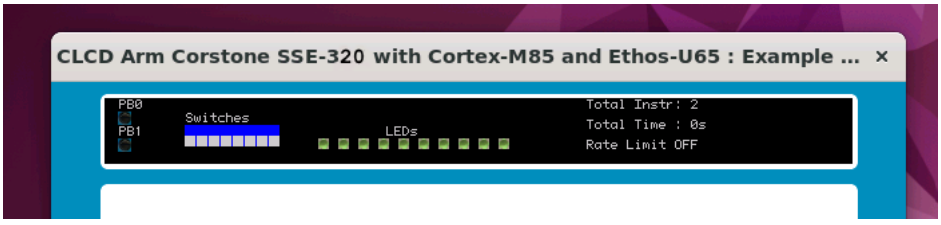
When an FVP starts, the CLCD window opens, representing the contents of the simulated color LCD frame buffer. It automatically resizes to match the horizontal and vertical resolution that is set in the CLCD peripheral registers.

MPS4 FVPs

The LEDs and MCC switches in the CLCD window for these FVPs correspond to the LEDs and switches on the physical board. They are controlled by the software that you load onto the FVP.

For information on how to use them, see User switches and user LEDs section in *Arm® MPS4 FPGA Prototyping Board Technical Reference Manual*.

Figure 2-1: MPS4 CLCD window in its default state at startup



Switches

Eight white boxes show the state of the User DIP switches.

These white boxes represent switches on the MPS4 hardware, which are mapped to register SWITCH(0x28) of System Control and I/O block. The switches are in the off position by default. To change its state, click in the area above or below a white box.

LEDs

Eight colored boxes indicate the state of the MPS4 User LEDs.

These colored boxes represent the red/yellow/green LEDs on the MPS4 hardware, which are mapped to register LED0(0x0) of System Control and I/O block.

Push Buttons

Two Push-Buttons PB0 and PB1. They are asserted when pressed and deasserted when released.

These represent push-buttons on the MPS4 hardware, which are mapped to register BUTTON(0x08) of System Control and I/O block.

Total Instr

A counter showing the total number of instructions executed.

Because the FVP models provide a Programmer's View (PV) of the system, the CLCD displays total instructions rather than total processor cycles. Timing might differ substantially from the hardware because:

- Bus fabric is simplified.
- Memory latencies are minimized.
- Cycle approximate processor and peripheral models are used.

In general, bus transaction timing is consistent with the hardware, but the timing of operations within the model is not accurate.

Total Time

A counter showing the total elapsed time, in seconds.

This time is wall clock time, not simulated time.

Rate Limit

A feature that disables or enables fast simulation.

Because the system model is highly optimized, your code might run faster than it would on real hardware. This effect might cause timing issues.

Rate Limit is disabled by default to favor simulation speed. Enable it to restrict simulation time so that it more closely matches real time.

To enable or disable Rate Limit, click the square button. You can also configure this option when instantiating the model with the rate_limit-enable visualization component parameter.

When you click the Total Instr item in the CLCD, the display toggles to show the following:

Instr/sec

The number of instructions that execute per second of wall clock time.

Perf Index

The ratio of real time to simulation time. The larger the ratio, the faster the simulation runs. If you enable the Rate Limit feature, the Perf Index approaches unity.

You can reset the simulation counters by resetting the model.

2.3 FPGA system control and I/O block

The MPS4 SSE-320 implements an FPGA system and I/O control block.

The following table lists the registers in FPGA system and I/O control block.

Table 2-1: FPGA system and I/O control register map

Offset	Name	Type	Reset	Description
0x000	LED0	RW	0x0000_0000	LED0
0x004	M85DBGCTR	**RAZ/WI**	0x0000_0000	Not modeled
0x008	BUTTON	R	0x0000_0000	BUTTON
0x00C	-	**RAZ/WI**	0x0000_0000	Reserved
0x010	CLK 1HZ	RW	0x0000_0000	CLK 1HZ
0x014	CLK 100Hz	RW	0x0000_0000	CLK 100HZ
0x018	COUNTER	RW	0x0000_0000	COUNTER
0x01C	PRESCALE	RW	0x0000_0000	PRESCALE
0x020	PSCNTR	RW	0x0000_0000	PSCNTR
0x024	-	**RAZ/WI**	0x0000_0000	Reserved
0x028	SWITCH	R	0x0000_000	SWITCH
0x04C	MISC	RW	0x0000_0000	MISC

2.3.1 LED0

Contains LED connections

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

0x000

Type

RW

Reset value

0x0000_0000

Bit descriptions

The following table shows the register bit assignments.

Table 2-2: LED0 bit descriptions

Bits	Name	Description	Type	Reset
[31:10]	Reserved	-	**RAZ/WI**	0x0
[9:0]	LED	LED conections	RW	0x0

2.3.2 BUTTON

Contains button connections

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

0x008

Type

R

Reset value

0x0000_0000

Bit descriptions

The following table shows the register bit assignments.

Table 2-3: BUTTON bit descriptions

Bits	Name	Description	Type	Reset
[31:2]	Reserved	-	**RAZ/WI**	0x000000
[1:0]	Buttons	?	R	0b0

2.3.3 CLK 1HZ

Contains LED connections

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

0x010

Type

RW

Reset value

0x0000_0000

Bit descriptions

The following table shows the register bit assignments.

Table 2-4: CLK 1HZ bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	CLK_1HZ	1Hz up counter	RW	0x0

2.3.4 CLK 100HZ

Contains LED connections

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

0x014

Type

RW

Reset value

0x0000_0000

Bit descriptions

The following table shows the register bit assignments.

Table 2-5: CLK 100HZ bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	CLK_100HZ	100Hz up counter	RW	0x0

2.3.5 COUNTER

Contains Cycle Up Counter which increments when 32-bit prescale counter equals zero and it automatically reloads

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

0x018

Type

RW

Reset value

0x0000_0000

Bit descriptions

The following table shows the register bit assignments.

Table 2-6: COUNTER bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	COUNTER	Cycle Up counter	RW	0x0

2.3.6 PRESCALE

Contains reload value for prescale counter

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

0x01C

Type

RW

Reset value

0x0000_0000

Bit descriptions

The following table shows the register bit assignments.

Table 2-7: PRESCALE bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	PRESCALE	Reload value for prescale counter	RW	0x0

2.3.7 PSCNTR

Contains current value of the prescale counter. The prescale counter is reloaded with PRESCALE after reaching 0

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

0x020

Type

RW

Reset value

0x0000_0000

Bit descriptions

The following table shows the register bit assignments.

Table 2-8: PSCNTR bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	PSCNTR	Current value of the prescale counter	RW	0x0

2.3.8 SWITCH

Contains LED connections

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

0x028

Type

R

Reset value

0x0000_0000

Bit descriptions

The following table shows the register bit assignments.

Table 2-9: LED0 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	Reserved	-	RAZ/WI	0x0
[7:0]	SWITCH	Switches	R	0x0

2.3.9 MISC

Contains miscellaneous control bits.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

0x04C

Type

RW

Reset value

0x0000_0000

Bit descriptions

The following table shows the register bit assignments.

Table 2-10: MISC bit descriptions

Bits	Name	Description	Type	Reset
[31:3]	Reserved	-	**RAZ/WI**	0x0
[2]	SHIELD1_SPI_nCS	?	RW	0x0
[1]	SHIELD0_SPI_nCS	?	RW	0x0
[0]	ADC_SPI_nCS	?	RW	0x0

2.4 Ethernet with VE FVPs

The VE FVPs have a virtual Ethernet component. This component is a model of the SMSC 91C111 Ethernet controller, and uses a TAP device to communicate with the network. By default, the Ethernet component is disabled.

Host requirements

Before you can use the Ethernet capability of VE FVPs, set up your host computer.

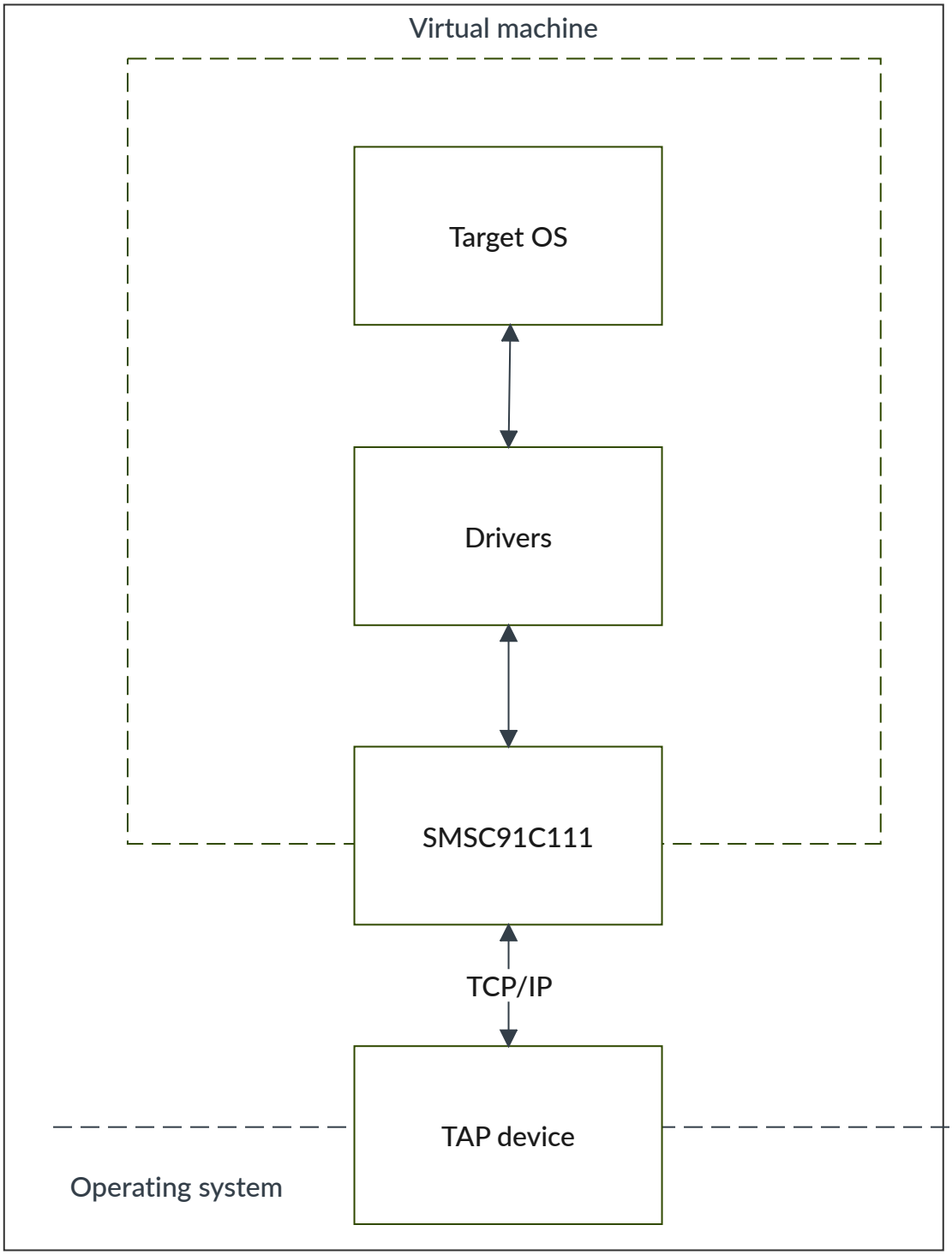
Target requirements

The VE FVPs include a software implementation of the SMSC 91C111 Ethernet controller. Your target OS must therefore include a driver for this specific device. To use the SMSC chip, configure the kernel. Linux supports the SMSC 91C111.

The configurable SMSC 91C111 component parameters are:

- enabled
- mac_address
- promiscuous

Figure 2-2: Model networking structure block



enabled

When the device is disabled, the kernel cannot detect the device.

To perform read and write operations on the TAP device, configure a HostBridge component. The HostBridge component is a virtual Programmer's View (PV) model. It acts as a networking gateway to exchange Ethernet packets with the TAP device on the host, and to forward packets to NIC models.

mac_address

There are two options for the mac_address parameter.

If a MAC address is not specified, when the simulator is run it takes the default MAC address, which is randomly generated. This random generation provides some degree of MAC address uniqueness when running models on multiple hosts on a local network.

promiscuous

The Ethernet component starts in promiscuous mode by default. In this mode, it receives all network traffic, even any not addressed to the device. Use this mode if you are using a single network device for multiple MAC addresses. Use this mode if, for example, you share the network card between your host OS and the VE FVP Ethernet component.

By default, the Ethernet device on the VE FVP has a randomly generated MAC address and starts in promiscuous mode.

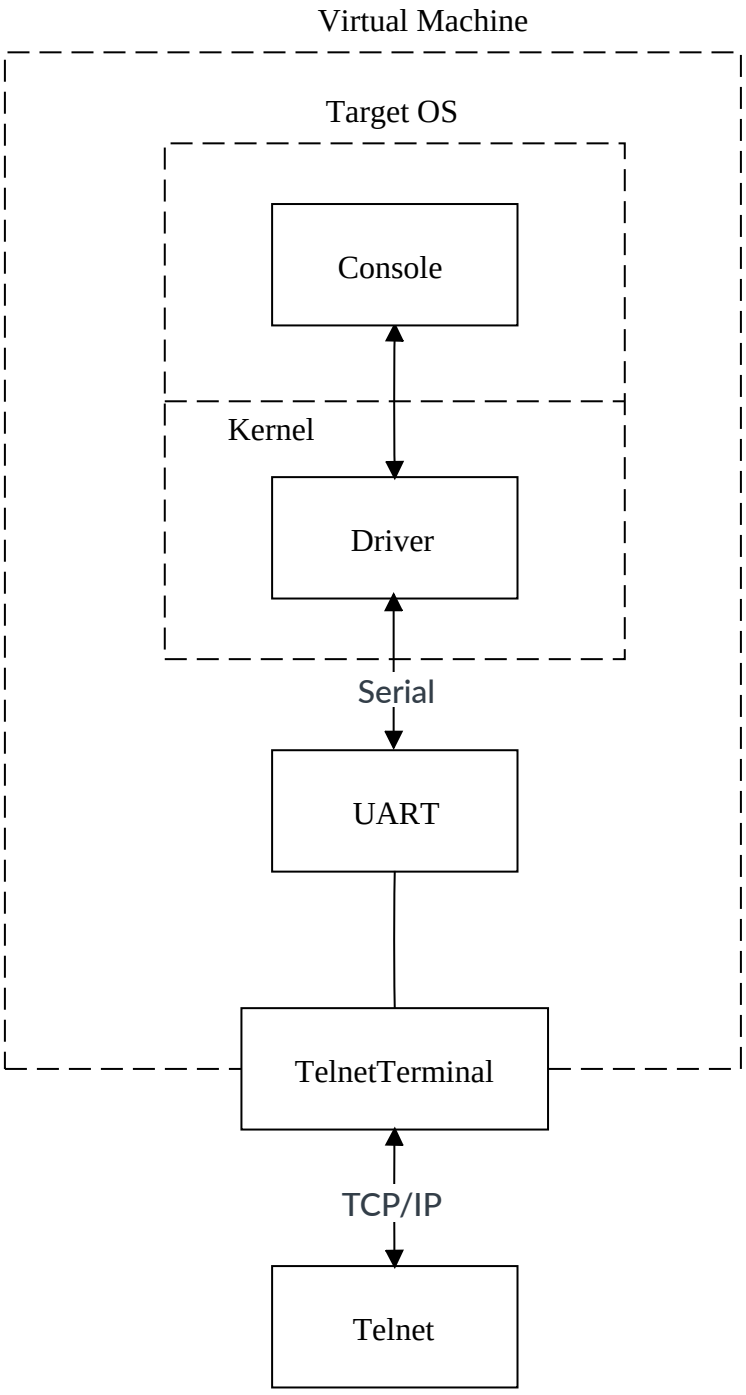
2.5 Using a terminal with a system model

The TelnetTerminal component is a virtual component that enables UART data to be transferred between a TCP/IP socket on the host and a serial port on the target

Using TelnetTerminal

The following figure shows a block diagram of one possible relationship between the target and host through the TelnetTerminal component. The TelnetTerminal block is what you configure when you define Terminal component parameters. The Virtual Machine is an FVP.

Figure 2-3: Terminal block diagram



On the target side, the console process that is invoked by your target OS relies on a suitable driver being present. Such drivers are normally part of the OS kernel. The driver passes serial data through a UART. The data is forwarded to the TelnetTerminal component. When the simulation is started and the TelnetTerminal component is enabled, the component opens a server (listening)

socket on a TCP/IP port. This is port 5000 by default. This port can be connected to by, for example, a Telnet process on the host.

When data becomes available on the network socket, the TelnetTerminal component buffers the data, which can then be read from SerialData.

If there is no connection to the network socket when the first data access is made, and the start_telnet parameter is true, a host Telnet session is started automatically. Prior to this first access, you can connect a client of your choice to the network socket.

If the connection between the TelnetTerminal component and the client is broken at any time, for example by closing a client Telnet session, the port is re-opened on the host, permitting you to make another client connection. This could have a different port number if the original one is no longer available.

The port number of a particular TelnetTerminal instance can be defined when your model system starts. The actual value of the port used by each TelnetTerminal is declared when it starts or restarts, and might not be the value that you specified if the port is already in use. If you are using Model Shell, the port numbers are displayed in the host window in which you started the model.


Note

Microsoft Windows version 8 and later, disable the Telnet client by default. If not enabled, follow these steps to enable it:

1. Select Start > Settings.
2. In the search box, type Turn Windows features on or off. The Windows Features dialog opens.
3. Select the Telnet Client check box and click OK. The installation might take several minutes to complete.

TelnetTerminal parameters

To set the parameters, the syntax to use in a configuration file or on the command line is:

motherboard.terminal_`x`.`parameter`=value

where `x` is the terminal identifier and can be 0, 1, 2, or 3.

You can start the TelnetTerminal component in either of the following modes, depending on the mode parameter:

telnet

In Telnet mode, the terminal component supports a subset of the RFC 854 protocol. This means that the terminal participates in negotiations between the host and client concerning what is and is not supported, but there is no flow control.

raw

In raw mode the byte stream passes unmodified between the host and the target. The terminal does not participate in initial capability negotiations between the host and client. Instead it acts as a TCP/IP port.

You can use this feature to directly connect to your target through the TelnetTerminal component. This permits a debugger connection, for example, to connect a gdb client to a gdbserver running on the target operating system.

The `terminal_command`parameter specifies the command line used to launch a terminal application and connect to the opened TCP port. The TelnetTerminal component replaces the keywords `%port` and `%title`, if specified, with the opened port number and component name, respectively. After replacing `%port` and `%title`, the command line is executed verbatim.

An empty string, which is the default, launches xterm on Linux or `telnet.exe` on Windows.

If you specify a non-empty string, it must include `%port`, but `%title` is optional.

For example:

```
fvp_mps2.telnetterminal0.terminal_command="putty.exe -telnet localhost %port"
```

3. SSE-320 FVP

To develop ahead of hardware availability and explore the design from a software perspective, the Corstone™ SSE-320 Fixed Virtual Platform (FVP) models much of the Arm IP in the Corstone™ SSE-320 design. It is a software model that approximates version r0p0 of the Corstone™ SSE-320 subsystem.

The FVP drives system architecture and software standardization. It is used with the Corstone™ SSE-320 software package which provides software and binaries of proprietary firmware that reduce the amount of work that is required for SoC development.

For instructions on how to set up and run the FVP, see Getting started with Fixed Virtual Platforms.

The FVP models the Corstone SSE-320 Subsystem including following IP components, though with some limitations:

- A single Arm® Cortex-M85 processor with MVE extension
- A single Arm® Ethos-U85 processor
- A single Arm® Mali™-C55 image signal processor
- Memory Protection Controller (MPC)
- Peripheral Protection Controller (PPC)
- Implementation Defined Attribution Unit (IDAU)
- Manager Security Controller (MSC)
- Arm® HDLCD video controller
- Arm® Lifecycle Manager (LCM)
- Arm® Key Management Unit (KMU)
- Arm® Security Alarm Manager (SAM)

The full description of components that are internal to the SSE-320 subsystem can be found in the *Arm® Corstone™ SSE-320 Example Subsystem Software Programmers Guide*.

The FVP has the following limitations:

- It does not model every component that SSE-320 describes. For example, it does not model the CoreSight™ SOC -600
- QSPI is not modeled. It has QSPI SRAM instead.
- Some components are partially modeled, and are defined as such in the following tables.

Additional information

- *Arm® Corstone™ SSE-320 Example Subsystem Software Programmers Guide*
- *Arm® MPS4 FPGA Prototyping Board Technical Reference Manual*
- *Arm® Mali™-C55 Image Signal Processor Technical Reference Manual*

- *Arm® Lifecycle Manager Specification*
- *Arm® Key Management Unit Specification*
- *Arm® Security Alarm Manager Specification*

3.1 SSE-320 FVP memory map

The memory map implementation aligns with SSE-320 Subsystem memory map. The memory map is expanded to show the supported MPS4 peripherals in the Expansion Manager Regions and their mapping.

Overview

This table outlines the main FVP memories and their positions within the memory map.

For information on the SSE-320 Subsystem peripherals, see the *Arm® Corstone™ SSE-320 Example Subsystem Software Programmers Guide*.

Security values do not define Privileged or Unprivileged accessibility. These are defined by the PPC, or by the register blocks that are mapped to each area. See lower-level details of each area for details.

For information on the SSE-320 Subsystem host system memory map, see *Arm® Corstone™ SSE-320 Example Subsystem Software Programmers Guide*.

The System Address Map defines the following values for accessibility:

S
: Secure Access

NS
: Non-secure Access

NSC
: Non-Secure Callable

The NSC values are defined through registers in the Secure Access Configuration registers.

Table 3-1: High-Level System Address Map

Row ID	Address	Size	Region	Alias	IDAU Region Values	Description
1	`0x0000_0000 - 0x0000_7FFF`	32KB	Code	7	Security: NS IDAUID: 0 NSC: 0	Instruction TCM
2	`0x0000_8000 - 0x00FF_FFFF`	15.968MB	Reserved	Reserved	-	-

Row ID	Address	Size	Region	Alias	IDAU Region Values	Description
3	0x0100_0000 - 0x011F_FFFF	2MB	Code	11	Security: NS IDAUID: 0 NSC: 0	FPGA SRAM
4	0x0120_0000 - 0x09FF_FFFF	142MB	Reserved	Reserved	-	-
5	0x0A00_0000 - 0x0A00_7FFF	32KB	Code	13	Security: NS IDAUID: 0 NSC: 0	DMA ITCM CPU0 S-AHB Instruction TCM Access
6	0x0A00_8000 - 0x0FFF_FFFF	95.968MB	Reserved	Reserved	-	-
7	0x1000_0000 -0x1000_7FFF	32KB	Code	1	Security: S IDAUID: 1 NSC: CODENSC	ITCM
8	0x1000_8000 - 0x10FF_FFFF	15.968MB	Reserved	Reserved	-	-
9	0x1100_0000 - 0x1101_FFFF	128KB	Code	-	Security: S IDAUID: 1 NSC: CODENSC	ROM
10	0x1101_0000 - 0x11FF_FFFF	15.936MB	Reserved	Reserved	-	-
11	0x1200_0000 - 0x121F_FFFF	2MB	Code	3	Security: S IDAUID: 1 NSC:	FPGA SRAM
12	0x1220_0000 - 0x19FF_FFFF	126MB	Reserved	Reserved	-	-
13	0x1A00_0000 - 0x1A00_7FFF	32KB	Code	5	Security: S IDAUID: 1 NSC: CODENSC	DMA ITCM
14	0x1A00_8000 - 0x1FFF_FFFF	95.968MB	Reserved	Reserved	-	-
15	0x2000_0000 - 0x2000_7FFF	32KB	SRAM	26	Security: NS IDAUID: 2 NSC: 0	DTCM
16	0x2000_8000 - 0x20FF_FFFF	15.968M	Reserved	Reserved	-	-

Row ID	Address	Size	Region	Alias	IDAU Region Values	Description
17	0x2100_0000 - 0x211F_FFFF	2MB	SRAM (VM0)	28	Security: NS IDAUID: 2 NSC: 0	Internal SRAM
18	0x2120_0000 - 0x213F_FFFF	2MB	SRAM (VM1)	29	Security: NS IDAUID: 2 NSC: 0	Internal SRAM
19	-		NA	-	-	-
20	-		NA	-	-	-
21	0x2400_0000 - 0x2400_7FFF	32KB	SRAM	32	Security: NS IDAUID: 2 NSC: 0	DMA DTCM
22	0x2400_8000 - 0x27FF_FFFF	63.968MB	63.968MB	Reserved	-	-
23	0x2800_0000 - 0x2FFF_FFFF	128MB	Flash Memory	34	Security: NS IDAUID: 2 NSC: 0	QSPI Flash
24	-	-	NA	-	-	-
25	-	-	NA	-	-	-
26	0x3000_0000 - 0x3000_7FFF	32KB	SRAM	15	Security: S IDAUID: 3 NSC: RAMNSC	DTCM
27	0x3000_8000 - 0x30FF_FFFF	15.968MB	Reserved	Reserved	-	-
28	0x3100_0000 - 0x311F_FFFF	2MB	SRAM (VM0)	17	Security: S IDAUID: 3 NSC: RAMNSC	Internal SRAM
29	0x3120_0000 - 0x313F_FFFF	2MB	SRAM (VM1)	18	Security: S IDAUID: 3 NSC: RAMNSC	Internal SRAM
30	0x3120_0000 - 0x31FF_FFFF	14MB	Reserved	Reserved	-	-
31	0x3200_0000 - 0x33FF_FFFF	32MB	Reserved	Reserved	-	-

Row ID	Address	Size	Region	Alias	IDAU Region Values	Description
32	0x3400_0000 - 0x3400_7FFF	32KB	SRAM	21	Security: S IDAUID: 3 NSC: RAMNSC	DMA DTCM
33	0x3400_8000 - 0x37FF_FFFF	63.968MB	Reserved	Reserved	-	-
34	0x3800_0000 - 0x3FFF_FFFF	128MB	Flash Memory	22	Security: S IDAUID: 3 NSC: RAMNSC	QSPI Flash
35	-	-	NA	-	-	-
36	-	-	NA	-	-	-
37	0x4000_0000 - 0x4000_FFFF	64KB	Peripherals	47	Security: NS IDAUID: 4 NSC: 0	Peripherals done
38	0x4001_0000 - 0x4001_FFFF	64KB	Private CPU	48	Security: NS IDAUID: 4 NSC: 0	Private CPU
39	0x4002_0000 - 0x4003_FFFF	128KB	Peripherals	49	Security: NS IDAUID: 4 NSC: 0	System Control Peripheral Region
40	0x4004_0000 - 0x400F_FFFF	768KB	Peripherals	/-	Security: NS IDAUID: 4 NSC: 0	Peripherals
41	0x4010_0000 - 0x47FF_FFFF	127MB	Peripheral Expansion	55	Security: NS IDAUID: 4 NSC: 0	Manager Peripheral Expansion Interface
42	0x4800_0000 - 0x4800_FFFF	64KB	Peripherals	56	Security: NS IDAUID: 4 NSC: 0	Peripheral Region
43	0x4801_0000 - 0x4801_FFFF	64KB	Private CPU	57	Security: NS IDAUID: 4 NSC: 0	Private CPU

Row ID	Address	Size	Region	Alias	IDAU Region Values	Description
44	0x4802_0000 - 0x4803_FFFF	128KB	System Control	58	Security: NS IDAUID: 4 NSC: 0	System Control Peripheral Region
45	0x4804_0000 - 0x480F_FFFF	768KB	Peripherals	-	Security: NS IDAUID: 4 NSC: RAMNSC	Peripherals
46	0x4810_0000 - 0x4FFF_FFFF	127MB	Peripheral Expansion	62	Security: NS IDAUID: 4 NSC: RAMNSC	Manager Peripheral Expansion Interface
47	0x5000_0000 - 0x5000_FFFF	64KB	Peripherals	37	Security: S IDAUID: 5 NSC: RAMNSC	Peripherals
48	0x5001_0000 - 0x5001_FFFF	64KB	Private CPU	38	Security: S IDAUID: 5 NSC: RAMNSC	Private CPU done
49	0x5002_0000 - 0x5003_FFFF	64KB	Peripherals	39	Security: S IDAUID: 5 NSC: RAMNSC	System Control Peripheral Region
50	0x5004_0000 - 0x5009_DFFF	375.99KB	Peripherals	-	Security: S IDAUID: 5 NSC: RAMNSC	Peripherals
51	0x5009_E000 - 0x5009_EFFF	4KB	KMU	-	Security: S IDAUID: 5 NSC: 0	KMU
52	0x5009_F000 - 0x5009_FFFF	4KB	Peripherals	-	Security: S IDAUID: 5 NSC: RAMNSC	Peripherals
53	0x500A_0000 - 0x500A_FFFF	64KB	LCM	-	Security: S IDAUID: 5 NSC: 0	LCM

Row ID	Address	Size	Region	Alias	IDAU Region Values	Description
54	0x500B_0000 - 0x500F_FFFF	64KB	Peripherals	-	Security: S IDAUID: 5 NSC: 0	Peripherals
55	0x5010_0000 - 0x57FF_FFFF	127MB	Peripheral expansion	41	Security: S IDAUID: 5 NSC: 0	Manager Peripheral Expansion Interface
56	0x5800_0000 - 0x5800_FFFF	64KB	Peripherals	42	Security: S IDAUID: 5 NSC: 0	Peripheral Region
57	0x5801_0000 - 0x5801_FFFF	64KB	Private CPU	43	Security: S IDAUID: 5 NSC: 0	Private CPU
58	0x5802_0000 - 0x5803_FFFF	128KB	System control	-	Security: S IDAUID: 5 NSC: 0	System control
59	0x5804_0000 - 0x580F_FFFF	768KB	Peripherals	-	Security: S IDAUID: 5 NSC: 0	Peripherals
62	0x5810_0000 - 0x5FFF_FFFF	127MB	Peripheral Expansion	46	Security: S IDAUID: 5 NSC: 0	Manager Peripheral Expansion Interface
63	0x6000_0000 - 0x6FFF_FFFF	256MB	Main Expansion	-	Security: NS IDAUID: 6 NSC: 0	DDR4
64	0x7000_0000 - 0x7FFF_FFFF	256MB	Main Expansion	-	Security: S IDAUID: 7 NSC: 0	DDR4 DDR4
65	0x8000_0000 - 0x8FFF_FFFF	256MB	Main Expansion	-	Security: NS IDAUID: 8 NSC: 0	DDR4

Row ID	Address	Size	Region	Alias	IDAU Region Values	Description
66	0x9000_0000 - 0x9FFF_FFFF	256MB	Main Expansion	-	Security: S IDAUID: 9 NSC: 0	DDR4
67	0xA000_0000 - 0xAFFF_FFFF	256MB	Main Expansion	-	Security: NS IDAUID: A NSC: 0	DDR4
68	0xB000_0000 - 0xBFFF_FFFF	256MB	Main Expansion	-	Security: S IDAUID: B NSC: 0	DDR4
69	0xC000_0000 - 0xCFFF_FFFF	256MB	Main Expansion	-	Security: NS IDAUID: 5C NSC: 0	DDR4
70	0xD000_0000 - 0xDFFF_FFFF	256MB	Main Expansion	-	Security: S IDAUID: D NSC: 0	DDR4
71	0xE000_0000 - 0xE00F_FFFF	1MB	PPB	-	Exempt	PPB - Not modeled
72	0xE010_0000 - 0xE01F_FFFF	127MB	Debug system	-	Security: NS IDAUID: E NSC: 0	Debug system - Not modeled
73	0xE020_0000 - 0xEFFF_FFFF	256MB	Peripheral Expansion	-	Security: NS IDAUID: F NSC: 0	Manager Peripheral Expansion Interface
74	0xF000_0000 - 0xF00F_FFFF	1MB	Reserved	-	Security: S IDAUID: F NSC: 0	Reserved
75	0xF010_0000 - 0xF01F_FFFF	1MB	Debug system	-	Security: S IDAUID: F NSC: 0	Debug system - Not modeled

Row ID	Address	Size	Region	Alias	IDAU Region Values	Description
76	0xF020_0000 - 0xFFFF_FFFF	254MB	Peripheral Expansion	-	Security: S IDAUID: F NSC: 0	Manager Peripheral Expansion Interface

3.2 SSE-320 Peripheral Region

The Corstone™ SSE-320 FVP includes peripherals that the software payload requires to run.

These peripherals are organized in the following layers:

Subsystem

The subsystem peripherals represent peripherals that are present on the SoC.

Board

The board peripherals represent peripherals that may be present on the board onto which the SoC is mounted. The SSE-320 board model is based on the Arm® MPS4 board.

For regions that are aliased to both Secure and Non-secure region, the final mapping of a peripheral in these regions to either Secure or Non-secure region is determined by Peripheral Protection Controller (PPC) that are programmed using Secure Access Configuration registers.

Peripherals implemented in these regions support 32-bit R/W accesses. Any Byte and Half word access results in **UNPREDICTABLE** behavior, unless otherwise stated.

The following table shows the memory map of the Peripheral Regions.

NS_PPC

Non-secure access only, gated by a PPC.

S_PPC

Secure access only, gated by a PPC.

S

Secure access only.

NS

Non-secure access only.

P

Privileged access only

UP

Unprivileged and privileged access allowed

P_PPC

Privileged access controlled by a PPC

Table 3-2: Peripheral Region Address Map

Row ID	Address	Size	Region name	Description	Alias with row ID	Security
1	0x4000_0000 - 0x4000_0FFF	4KB	Reserved	-	23	NS_PPC, P_PPC
2	0x4000_1000 - 0x4000_1FFF	4KB	Reserved	-	24	
3	0x4000_2000 - 0x4000_3FFF	8KB	DMA	DMA Configuration interface	25	NS_PPC, P_PPC
4	0x4000_4000 - 0x4000_7FFF	16KB	NPU0	If NPUNUM >0, NPU0 Configuration interface. Reserved if NPUNUM =0.	26	NS_PPC, P_PPC
5	-	-	NA	-	-	-
6	-	-	NA	-	-	-
7	-	-	NA	-	-	-
8	0x4000_8000 - 0x4008_FFFF	-	Reserved	-	-	-
9	0x4004_0000 - 0x4007_FFFF	-	Reserved	Reserved	-	-
10	0x4008_0000 - 0x4008_0FFF	4KB	NSACFG	Non-secure Access Configuration Register Block.	-	NS_PPC, P, P_PPC
11	0x4008_1000 - 0x4008_FFFF	-	Reserved	-	-	-
12	0x4009_0000 - 0x4009_3FFF	16KB	Reserved	-	-	NS
13	0x4009_4000 - 0x400F_FFFF	-	Reserved	-	-	-
14	0x4800_0000 - 0x4800_0FFF	4KB	TIMER0	Timer 0.	43	NS_PPC, P_PPC
15	0x4800_1000 - 0x4800_1FFF	4KB	TIMER1	Timer 1.	44	
16	0x4800_2000 - 0x4800_2FFF	4KB	TIMER2	Timer 2.	45	
17	0x4800_3000 - 0x4800_3FFF	4KB	TIMER3	Timer 3.	46	
18	0x4800_F000 - 0x4800_FFFF	4KB	Non-secure SDC 600	SDC-600 Internal APBCOM. If SDC-600 does not exist, this area is Reserved.	47	
19	0x4804_0000 - 0x4804_0FFF	4KB	NSWDCTRL	Non-secure Watchdog Control Frame.	-	NS_PPC, P_PPC, P
20	0x4804_1000 - 0x4804_1FFF	4KB	NSWDREF	Non-secure Watchdog Refresh Frame.	-	
21	0x4804_2000 - 0x4804_FFFF	-	Reserved	-	-	-
22	0x4805_0000 - 0x480F_FFFF	-	Reserved	-	-	-

Row ID	Address	Size	Region name	Description	Alias with row ID	Security
23	0x5000_0000 - 0x5000_0FFF	4KB	Reserved	-	-	-
24	0x5000_1000 - 0x5000_1FFF	4KB	Reserved	-	-	-
25	0x5000_2000 - 0x5000_3FFF	8KB	DMA	DMA Configuration interface.	3	S, UP
26	0x5000_4000 - 0x5000_7FFF	16KB	NPU0	If NPUNUM >0, NPU0 Configuration interface. Reserved if NPUNUM =0.	4	S_PPC, P_PPC
27	-	-	NA	-	-	-
28	-	-	NA	-	-	-
29	-	-	NA	-	-	-
30	0x5000_8000 - 0x5000_FFFF	-	Reserved	Reserved (**RAZ/WI**)	-	-
31	0x5004_0000 - 0x5007_FFFF	-	Reserved	Reserved	-	-
32	0x5008_0000 - 0x5008_0FFF	4KB	SACFG	Secure Access Configuration Register Block.	-	NS_PPC, P_PPC, P
33	0x5008_1000 - 0x5008_2FFF	-	Reserved	-	-	-
34	0x5008_3000 - 0x5008_3FFF	4KB	VM0MPC	VM0 Memory Protection Controller.	-	NS_PPC, P_PPC, P
35	0x5008_4000 - 0x5008_4FFF	4KB	VM1MPC	VM1 Memory Protection Controller.	-	
36	0x5008_5000 - 0x5008_5FFF	4KB	Reserved	-	-	-
37	0x5008_6000 - 0x5008_6FFF	4KB	Reserved	-	-	-
38	0x5008_7000 - 0x5009_DFFF	-	Reserved	-	-	-
39	0x5009_E000 - 0x5009_EFFF	4KB	KMU	When LCM_KMU_SAM_PRESENT = 0 this is Reserved.	-	S_PPC, P_PPC
40	0x5009_F000 - 0x5009_FFFF	4KB;	Reserved	-	-	-
41	0x500A_0000 - 0x500A_FFFF	64KB	LCM	When LCM_KMU_SAM_PRESENT = 0 this is Reserved.	-	S_PPC, P_PPC
42	0x500B_0000 - 0x57FF_FFFF	-	Reserved	-	-	-
43	0x5800_0000 - 0x5800_0FFF	4KB	TIMER0	Timer 0.	14	S_PPC, P_PPC
44	0x5800_1000 - 0x5800_1FFF	4KB	TIMER1	Timer 1.	15	
45	0x5800_2000 - 0x5800_2FFF	4KB	TIMER2	Timer 2.	16	
46	0x5800_3000 - 0x5800_3FFF	4KB	TIMER3	Timer 3.	17	

Row ID	Address	Size	Region name	Description	Alias with row ID	Security
47	0x5800_F000 - 0x5800_FFFF	4KB;	Secure SDC600	SDC-600 Internal APBCOM. If SDC-600 does not exist, this area is Reserved (**RAZ/WI**)	18	
48	0x5804_0000 - 0x5804_0FFF	4KB	SWDCTRL	Secure Watchdog Control Frame.	-	S_PPC, P_PPC
49	0x5804_1000 - 0x5804_1FFF	4KB	SWDREF	Secure Watchdog Refresh Frame.	-	
50	0x5804_2000 - 0x5804_2FFF	4KB	SAM	When LCM_KMU_SAM_PRESENT = 0, this is Reserved.	-	S_PPC, P_PPC
51	0x5804_3000 - 0x5804_FFFF	-	Reserved	-	-	-
52	0x5805_0000 - 0x580F_FFFF	-	Reserved	-	-	-

3.3 SSE-320 board expansion port and system maps

All FPGA peripherals are mapped to four regions of the memory map.

For information on the SSE-320 peripherals, see the *Arm® Corstone™ SSE-320 Example Subsystem Software Programmers Guide*.

Expansion system peripherals

The following table lists the addresses and interfaces of the four regions of the memory map:

Table 3-3: Expansion System Peripherals mapping

Address	Region	Description
0x4010_0000 - 0x47FF_FFFF	Non-Secure Low Latency	Table 3-4: MSTEXPPILL Non-secure low-latency peripheral map on page 39
0x4810_2000 - 0x4FFF_FFFF	Non-Secure High Latency	Table 3-5: MSTEXPPIHL Non-secure high-latency peripheral map on page 40
0x5010_0000 - 0x57FF_FFFF	Secure Low Latency	Table 3-6: MSTEXPPILL Secure low-latency peripheral map on page 41
0x5810_2000 - 0x5FFF_FFFF	Secure High Latency	Table 3-7: MSTEXPPIHL Secure high-latency peripheral map on page 41

To support TrustZone® for Armv8-M and allow software to map these peripherals to Secure or NonSecure address space, all peripherals are mapped twice and either an APB PPC or an AHB PPC gates the access to these peripherals.

Table 3-4: MSTEXPPILL Non-secure low-latency peripheral map

ROW ID	Address	Size	Description	Details
1	0x4010_0000 - 0X4010_0FFF	4KB	GPIO 0	CMSDK GPIO
2	0x4010_1000 - 0x4010_1FFF	4KB	GPIO 1	CMSDK GPIO
3	0x4010_2000 - 0x4010_2FFF	4KB	GPIO 2	CMSDK GPIO

ROW ID	Address	Size	Description	Details
4	0x4010_3000 - 0x4010_3FFF	4KB	GPIO 3	CMSDK GPIO
5	0x4031_0000 - 0x403_10FFF	4KB	HDLCD	PL370 HDLCD
6	0x4040_0000 - 0x404F_FFFF	1MB	Ethernet	SMSC 91C111
7	0x4050_0000 - 0x405F_FFFF	1MB	USB	Not modeled
8	0x4070_0000 - 0x4070_0FFF	4KB	Timing Adapter APB 0 – FPGA SRAM	Not modeled
9	0x4070_1000 - 0x4070_1FFF	4KB	Timing Adapter APB 1 - QSPI	Not modeled
10	0x4070_2000 - 0x4070_2FFF	4KB	Timing Adapter APB 2 – DDR4	Not modeled

For details of HDLCD, see the System IP components section of the *Fast Models Reference Guide*.

Table 3-5: MSTEXPPIHL Non-secure high-latency peripheral map

ROW ID	Address	Size	Description	Details
1	0x4810_2000 - 0x4810_2FFF	4KB	FPGA - PL022 (SPI ADC)	Not modeled
2	0x4810_3000 - 0x4810_3FFF	4KB	FPGA - PL022 (SPI Shield0)	Not modeled
3	0x4810_4000 - 0x4810_4FFF	4KB	FPGA - PL022 (SPI Shield1)	Not modeled
4	0x4810_5000 - 0x4810_5FFF	4KB	SBCon (I2C - Shield0)	Not modeled
5	0x4810_6000 - 0x4810_6FFF	4KB	SBCon (I2C – Shield1)	Not modeled
6	0x4810_8000 - 0x4810_8FFF	4KB	FPGA - SBCon I2C (DDR4 EEPROM)	Not modeled
7	0x4810_9000 - 0x4810_9FFF	4KB	FPGA - SBCon I2C (Audio Conf)	Not modeled
8	0x4820_0000 - 0x4820_0FFF	4KB	FPGA - SCC registers	Modeled
9	0x4820_1000 - 0x4820_1FFF	4KB	FPGA - I2S (Audio)	Partially modeled
10	0x4820_2000 - 0x4820_2FFF	4KB	FPGA - IO (System Ctrl + I/O)	Modeled For details, see FPGA system control and I/O block
11	0x4820_3000 - 0x4820_3FFF	4KB	UART0 - FPGA_UART0	CMSDK UART
12	0x4820_4000 - 0x4820_4FFF	4KB	UART1 - FPGA_UART1	CMSDK UART
13	0x4820_5000 - 0x4820_5FFF	4KB	UART2 - FPGA_UART2	CMSDK UART
14	0x4820_6000 - 0x4820_6FFF	4KB	UART3 - UART Shield 0	CMSDK UART
15	0x4820_7000 - 0x4820_7FFF	4KB	UART4 - UART Shield 1	CMSDK UART
16	0x4820_8000 - 0x4820_8FFF	4KB	UART5 - FPGA_UART3	CMSDK UART

ROW ID	Address	Size	Description	Details
17	0x4820_B000 - 0x4820_BFFF	4KB	RTC	PL031 RTC
18	0x4830_0000 - 0x4830_FFFF	1MB	ISP C55	Modeled
19	0x4840_0000 - 0x4840_0FFF	4KB	ISP Camera Config	Virtual

Table 3-6: MSTEXPPILL Secure low-latency peripheral map

ROW ID	Address	Size	Description	Details
1	0x5010_0000 - 0x5010_0FFF	4KB	GPIO 0	CMSDK GPIO
2	0x5010_1000 - 0x5010_1FFF	4KB	GPIO 1	CMSDK GPIO
3	0x5010_2000 - 0x5010_2FFF	4KB	GPIO 2	CMSDK GPIO
4	0x5010_3000 - 0x5010_3FFF	4KB	GPIO 3	CMSDK GPIO
5	0x5031_0000 - 0x503_10FFF	4KB	HDLCD	PL370 HDLCD
6	0x5040_0000 - 0x504F_FFFF	1MB	Ethernet	SMSC 91C111
7	0x5050_0000 - 0x505F_FFFF	1MB	USB	Not modeled
8	0x5070_0000 - 0x5070_0FFF	4KB	Timing Adapter APB 0 – FPGA SRAM	Not modeled
9	0x5070_1000 - 0x5070_1FFF	4KB	Timing Adapter APB 1 - QSPI	Not modeled
10	0x5070_2000 - 0x5070_2FFF	4KB	Timing Adapter APB 2 – DDR4	Not modeled

Table 3-7: MSTEXPPIHL Secure high-latency peripheral map

ROW ID	Address	Size	Description	Details
1	0x5810_2000 - 0x5810_2FFF	4KB	FPGA - PL022 (SPI ADC)	Not modeled
2	0x5810_3000 - 0x5810_3FFF	4KB	FPGA - PL022 (SPI Shield0)	Not modeled
3	0x5810_4000 - 0x5810_4FFF	4KB	FPGA - PL022 (SPI Shield1)	Not modeled
4	0x5810_5000 - 0x5810_5FFF	4KB	SBCon (I2C - Shield0)	Not modeled
5	0x5810_6000 - 0x5810_6FFF	4KB	SBCon (I2C – Shield1)	Not modeled
6	0x5810_8000 - 0x5810_8FFF	4KB	FPGA - SBCon I2C (DDR4 EEPROM)	Not modeled
7	0x5810_9000 - 0x5810_9FFF	4KB	FPGA - SBCon I2C (Audio Conf)	Not modeled
8	0x5820_0000 - 0x5820_0FFF	4KB	FPGA - SCC registers	Modeled
9	0x5820_1000 - 0x5820_1FFF	4KB	FPGA - I2S (Audio)	Partially modeled
10	0x5820_2000 - 0x5820_2FFF	4KB	FPGA - IO (System Ctrl + I/O)	Modeled
11	0x5820_3000 - 0x5820_3FFF	4KB	UART0 - FPGA_UART0	CMSDK UART
12	0x5820_4000 - 0x5820_4FFF	4KB	UART1 - FPGA_UART1	CMSDK UART
13	0x5820_5000 - 0x5820_5FFF	4KB	UART2 - FPGA_UART2	CMSDK UART
14	0x5820_6000 - 0x5820_6FFF	4KB	UART3 - UART Shield 0	CMSDK UART
15	0x5820_7000 - 0x5820_\7FFF	4KB	UART4 - UART Shield 1	CMSDK UART
16	0x5820_8000 - 0x5820_8FFF	4KB	UART5 - FPGA_UART3	CMSDK UART
17	0x5820_B000 - 0x5820_BFFF	4KB	RTC	PL031 RTC
18	0x5830_0000 - 0x5830_FFFF	1MB	ISP C55	Modeled
19	0x5840_0000 - 0x5840_0FFF	4KB	ISP Camera Config	Virtual

3.4 SSE-320 interrupt maps

The interrupts in the FPGA Subsystem extend the SSE-320 Interrupt map by adding to the expansion area.

Table 3-8: Interrupt map at the board layer

Interrupt input for CPU0	Interrupt source for CPU0	EWIC support
NMI	Combined Secure System Watchdog, SLOWCLK Watchdog, . CPU0EXPNMI and NMI of Security Alarm Manager When LCM_KMU_SAM_PRESENT = 0, no NMI of SAM present.	Yes
IRQ[0]	Non-secure watchdog reset request	Yes
IRQ[1]	Non-secure watchdog interrupt	Yes
IRQ[2]	SLOWCLK Timer	Yes
IRQ[3]	Timer 0	Yes
IRQ[4]	Timer 1	Yes
IRQ[5]	Timer 2	Yes
IRQ[6]	Reserved	-
IRQ[7]	Reserved	-
IRQ[8]	Reserved	-
IRQ[9]	MPC Combined (Secure)	Yes
IRQ[10]	PPC Combined (Secure)	Yes
IRQ[11]	MSC Combined (Secure)	Yes
IRQ[12]	Bridge Error Combined interrupt (Secure)	Yes
IRQ[13]	Reserved	-
IRQ[14]	PPU combined	Yes
IRQ[15]	SDC-600. Reserved if SDC-600 does not exist	Yes
IRQ[16]	NPU0	Yes
IRQ[17]	Reserved	-
IRQ[18]	Reserved	-
IRQ[19]	Reserved	-
IRQ[20]	Key Management Unit When LCM_KMU_SAM_PRESENT = 0, this IRQ is Reserved	Yes
IRQ[23:21]	Reserved	-
IRQ[24]	DMA (Combined Secure)	-
IRQ[25]	DMA (Combined Non-secure)	-
IRQ[26]	DMA (Security violation)	-
IRQ[27]	Timer 3 AON	Yes
IRQ[28]	CPU0CTIIRQ0 (local CPU CTI only)	No
IRQ[29]	CPU0CTIIRQ1 (local CPU CTI only)	No
IRQ[30]	Critical Severity Fault Interrupt of Security Alarm Manager. When LCM_KMU_SAM_PRESENT = 0, this IRQ is Reserved.	Yes
IRQ[31]	Severity Fault Interrupt of Security Alarm Manager. When LCM_KMU_SAM_PRESENT = 0, this IRQ is Reserved.	Yes

Interrupt input for CPU0	Interrupt source for CPU0	EWIC support
IRQ[33]	UART 0 Receive Interrupt	No
IRQ[34]	UART 0 Transmit Interrupt	No
IRQ[35]	UART 1 Receive Interrupt	No
IRQ[36]	UART 1 Transmit Interrupt	No
IRQ[37]	UART 2 Receive Interrupt	No
IRQ[38]	UART 2 Transmit Interrupt	No
IRQ[39]	UART 3 Receive Interrupt	No
IRQ[40]	UART 3 Transmit Interrupt	No
IRQ[48]	UART Overflow (0, 1, 2, 3, 4 & 5)	No
IRQ[49]	Ethernet	No
IRQ[57]	DMA Channel 0	No
IRQ[58]	DMA Channel 1	No
IRQ[69]	GPIO 0 Combined Interrupt	No
IRQ[70]	GPIO 1 Combined Interrupt	No
IRQ[71]	GPIO 2 Combined Interrupt	No
IRQ[72]	GPIO 3 Combined Interrupt	No
IRQ[125]	UART 5 Receive Interrupt	No
IRQ[126]	UART 5 Transmit Interrupt	No
IRQ[128]	RTC	No
IRQ[132]	ISP	No
IRQ[133]	HDLCD	No
IRQ[224]	VSI 0 Interrupt	No
IRQ[225]	VSI 1 Interrupt	No
IRQ[226]	VSI 2 Interrupt	No
IRQ[227]	VSI 3 Interrupt	No
IRQ[228]	VSI 4 Interrupt	No
IRQ[229]	VSI 5 Interrupt	No
IRQ[230]	VSI 6 Interrupt	No
IRQ[231]	VSI 7 Interrupt	No

3.5 SSE-320 FVP peripheral protection controller expansion map

The SSE-320 FVP implements secure access configuration registers that control security and privileged accesses to peripherals which are connected to the Peripheral Protection Controller (PPC).

For more details of registers, see *Arm® Corstone™ SSE-320 Example Subsystem Software Programmers Guide*.

Table 3-9: Secure access configuration registers - PPC bits

Bit	MAIN_PPCEXP0 (AHB0)	MAIN_PPCEXP1 (AHB1)	PERIPH_PPCEXP0 (APB0)	PERIPH_PPCEXP1 (APB1)	PERIPH_PPCEXP2 (APB2)
0	GPIO 0	-	Timing adapter APB 0	SBCon I2C (touch screen)	FPGA - SCC registers
1	GPIO 1	-	Timing adapter APB 1	SBCon I2C (audio conf)	FPGA - I2S (audio)
2	GPIO 2	-	Timing adapter APB 2	FPGA PL022 (SPI2 for ADC)	FPGA – GPIO (System Ctrl + I/O)
3	GPIO 3	-	-	FPGA PL022 (SPI Shield0)	UART0
4	-	-	-	FPGA PL022 (SPI Shield1)	UART1
5	-	-	-	FPGA SBCon (I2C Shield0)	UART2
6	-	-	-	FPGA SBCon (I2C Shield1)	UART3
7	HDLCD	-	-	-	UART4
8	USB and ethernet	-	-	FPGA – SBCon I2C (DDR4 EPROM)	UART5
9	-	-	-	-	-
10	-	-	-	-	-
11	-	-	-	-	RTC
12	-	-	-	-	-
13	-	-	-	-	-
14	-	-	-	-	-
15	-	-	-	-	-

The SECPPCINTSTAT, Secure PPC Interrupt status register allows software to determine the source of the interrupt. When access violations occur on any PPC, a level interrupt is raised through a combined interrupt that is then sent to the CPU0.

The following table displays the relevant bit of the SECPPCINTSTAT register for each PPC interrupt.

Table 3-10: SECPPCINTSTAT bits

\-	MAIN_PPCEXP0 (APB0)	MAIN_PPCEXP1 (AHB1)	PERIPH_PPCEXP0 (AHB0)	PERIPH_PPCEXP1 (APB1)	PERIPH_PPCEXP2 (APB2)
SECPPCINTSTAT bit	20	21	4	5	6

3.6 SSE-320 memory components

The SSE-320 FVP includes the following memory components and their security access is controlled by the MPC.

Table 3-11: Memory components

Name	Non-secure address range	Secure alias	Size
SRAM	`0x0100_0000 - 0x011F_FFFF`	`0x1200_0000 - 0x121F_FFFF`	2MB

Name	Non-secure address range	Secure alias	Size
QSPI SRAM	0x2800_0000 - 0x2FFF_FFFF	0x3800_0000 - 0x3FFF_FFFF	128MB

Table 3-12: DDR regions

Row ID	Address range	IDAU region security	Size	Description
1	0x6000_0000 - 0x6FFF_FFFF	NS	256MB	DDR0
2	0x7000_0000 - 0x7FFF_FFFF	S	256MB	DDR1
3	0x8000_0000 - 0x8FFF_FFFF	NS	256MB	DDR2
4	0x9000_0000 - 0x9FFF_FFFF	S	256MB	DDR3
5	0xA000_0000 - 0xAFFF_FFFF	NS	256MB	DDR4
6	0xB000_0000 - 0xBFFF_FFFF	S	256MB	DDR5
7	0xC000_0000 - 0xCFFF_FFFF	NS	256MB	DDR6
8	0xD000_0000 - 0xDFFF_FFFF	S	256MB	DDR7

3.7 SSE-320 virtual components

The SSE-320 FVP includes the virtual components.

Virtual interfaces

The Virtual Interfaces in Arm FVPs implement various CPU peripherals that allow to use external resources for stimulating the firmware application.

- The Virtual Input/Output (VIO) interface controls simple binary state I/O such as LED and switches.
- The Virtual Streaming Interface (VSI) provides up to 8 channels for data streaming.
- The Virtual Socket Interface (VSocket) connects the user application to BSD sockets on the Host computer for IP network connectivity.

For more details, see https://arm-software.github.io/AVH/main/simulation/html/group__arm__cmvp.html.

Table 3-13: VSI/VIO/VSOCKET Memory Map

Row ID	Address range	Size	Description
1	0x4FF0_0000 - 0x4FF0_FFFF	64KB	ARM_VSI0_BASE
2	0x4FF1_0000 - 0x4FF1_FFFF	64KB	ARM_VSI1_BASE
3	0x4FF2_0000 - 0x4FF2_FFFF	64KB	ARM_VSI2_BASE
4	0x4FF3_0000 - 0x4FF3_FFFF	64KB	ARM_VSI3_BASE
5	0x4FF0_0000 - 0x4FF4_FFFF	64KB	ARM_VSI4_BASE
6	0x4FF4_0000 - 0x4FF5_FFFF	64KB	ARM_VSI5_BASE
7	0x4FF5_0000 - 0x4FF6_FFFF	64KB	ARM_VSI6_BASE
8	0x4FF6_0000 - 0x4FF7_FFFF	64KB	ARM_VSI7_BASE
9	0x4FEE_0000 - 0x4FEE_FFFF	64KB	VSOCKET

Row ID	Address range	Size	Description
10	`0x4FEF_0000` - `0x4EFF_FFFF`	64KB	VIO

Integrated Signal Processor - ISP C55

Virtual Camera emulator Start Register `0x4830_0000` (uint8) needs to be set to 1 to start camera and ISP and set to 0 to stop camera and ISP.

Proprietary Notice

This document is protected by copyright and other related rights and the use or implementation of the information contained in this document may be protected by one or more patents or pending patent applications. No part of this document may be reproduced in any form by any means without the express prior written permission of Arm Limited ("Arm"). No license, express or implied, by estoppel or otherwise to any intellectual property rights is granted by this document unless specifically stated.

Your access to the information in this document is conditional upon your acceptance that you will not use or permit others to use the information for the purposes of determining whether the subject matter of this document infringes any third party patents.

The content of this document is informational only. Any solutions presented herein are subject to changing conditions, information, scope, and data. This document was produced using reasonable efforts based on information available as of the date of issue of this document. The scope of information in this document may exceed that which Arm is required to provide, and such additional information is merely intended to further assist the recipient and does not represent Arm's view of the scope of its obligations. You acknowledge and agree that you possess the necessary expertise in system security and functional safety and that you shall be solely responsible for compliance with all legal, regulatory, safety and security related requirements concerning your products, notwithstanding any information or support that may be provided by Arm herein. In addition, you are responsible for any applications which are used in conjunction with any Arm technology described in this document, and to minimize risks, adequate design and operating safeguards should be provided for by you.

This document may include technical inaccuracies or typographical errors. THIS DOCUMENT IS PROVIDED "AS IS". ARM PROVIDES NO REPRESENTATIONS AND NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENT. For the avoidance of doubt, Arm makes no representation with respect to, and has undertaken no analysis to identify or understand the scope and content of, any patents, copyrights, trade secrets, trademarks, or other rights.

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL ARM BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF ARM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Reference by Arm to any third party's products or services within this document is not an express or implied approval or endorsement of the use thereof.

This document consists solely of commercial items. You shall be responsible for ensuring that any permitted use, duplication, or disclosure of this document complies fully with any relevant

export laws and regulations to assure that this document or any portion thereof is not exported, directly or indirectly, in violation of such export laws. Use of the word “partner” in reference to Arm’s customers is not intended to create or refer to any partnership relationship with any other company. Arm may make changes to this document at any time and without notice.

This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of this document shall prevail.

The validity, construction and performance of this notice shall be governed by English Law.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

PRE-1121-V1.0

Product and document information

Read the information in these sections to understand the release status of the product and documentation, and the conventions used in Arm documents.

Product status

All products and services provided by Arm require deliverables to be prepared and made available at different levels of completeness. The information in this document indicates the appropriate level of completeness for the associated deliverables.

Product completeness status

The information in this document is Final, that is for a developed product.

Product revision status

The r0p0 identifier indicates the revision status of the product described in this manual, where:

r*x* Identifies the major revision of the product.
p*y* Identifies the minor revision or modification status of the product.

Revision history

These sections can help you understand how the document has changed over time.

Document release information

The Document history table gives the issue number and the released date for each released issue of this document.

Document history

Issue	Date	Confidentiality	Change
0000-01	27 September 2024	Non-Confidential	Initial release

Change history

The Change history tables describe the technical changes between released issues of this document in reverse order. Issue numbers match the revision history in Document release information on page 49.

Table 2: Issue 0000-01

Change	Location
Initial issue of the document	-

Conventions

The following subsections describe conventions used in Arm documents.

Glossary

The Arm Glossary is a list of terms used in Arm documentation, together with definitions for those terms. The Arm Glossary does not contain terms that are industry standard unless the Arm meaning differs from the generally accepted meaning.

See the Arm Glossary for more information: developer.arm.com/glossary.

Typographic conventions

Arm documentation uses typographical conventions to convey specific meaning.

Convention	Use
italic	Citations.
bold	Terms in descriptive lists, where appropriate.
`monospace`	Text that you can enter at the keyboard, such as commands, file and program names, and source code.
`monospace underline`	A permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.
`<and>`	Encloses replaceable terms for assembler syntax where they appear in code or code fragments. For example: `MRC p15, 0, <Rd>, <CRn>, <CRm>, <Opcode_2>`
SMALL CAPITALS	Terms that have specific technical meanings as defined in the *Arm® Glossary*. For example, **IMPLEMENTATION DEFINED**, **IMPLEMENTATION SPECIFIC**, **UNKNOWN**, and **UNPREDICTABLE**.

We recommend the following. If you do not follow these recommendations your system might not work.

Your system requires the following. If you do not follow these requirements your system will not work.

Danger
You are at risk of causing permanent damage to your system or your equipment, or of harming yourself.

Note
This information is important and needs your attention.

Tip
This information might help you perform a task in an easier, better, or faster way.

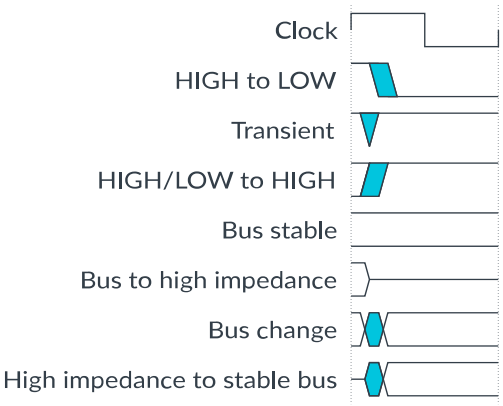
Remember
This information reminds you of something important relating to the current content.

Timing diagrams

The following figure explains the components used in timing diagrams. Variations, when they occur, have clear labels. You must not assume any timing information that is not explicit in the diagrams.

Shaded bus and signal areas are undefined, so the bus or signal can assume any value within the shaded area at that time. The actual level is unimportant and does not affect normal operation.

Figure 1: Key to timing diagram conventions



Signals

The signal conventions are:

Signal level

The level of an asserted signal depends on whether the signal is active-HIGH or active-LOW. Asserted means:

- HIGH for active-HIGH signals.
- LOW for active-LOW signals.

Lowercase n

At the start or end of a signal name, n denotes an active-LOW signal.

Useful resources

This document contains information that is specific to this product. See the following resources for other useful information.

Access to Arm documents depends on their confidentiality:

- Non-Confidential documents are available at developer.arm.com/documentation. Each document link in the following tables goes to the online version of the document.
- Confidential documents are available to licensees only through the product package.

Arm product resources	Document ID	Confidentiality
Arm® Corstone™ SSE-320 Example Subsystem Software Programmers Guide	109759	Non-Confidential
Arm® MPS4 FPGA Prototyping Board Technical Reference Manual	102577	Non-Confidential
Arm® Mali™-C55 Image Signal Processor Technical Reference Manual	102564	Confidential
Fast Models Reference Guide	100964	Non-Confidential

Arm architecture and specifications	Document ID	Confidentiality
Arm® Key Management Unit Specification	107715	Non-Confidential
Arm® Lifecycle Manager Specification	107616	Non-Confidential
Arm® Security Alarm Manager Specification	107716	Non-Confidential