# 25. On-Chip Memory (RAM and ROM) Core

## 25.1. Core Overview

Intel FPGAs include on-chip memory blocks that can be used as RAM or ROM in Platform Designer systems. On-chip memory has the following benefits for Platform Designer systems:

- On-chip memory has fast access time, compared to off-chip memory.
- Platform Designer automatically instantiates on-chip memory inside the Platform Designer system, so you do not have to make any manual connections.
- Certain memory blocks can have initialized contents when the FPGA powers up. This feature is useful, for example, for storing data constants or processor boot code.
- On-chip memories support dual port accesses, allowing two host to access the same memory concurrently.

## 25.2. Component-Level Design for On-Chip Memory

In Platform Designer you instantiate on-chip memory by clicking On-chip Memory (RAM or ROM) from the list of available components. The configuration window for the On-chip Memory (RAM or ROM) component has the following options: **Memory type**, **Size**, and **Read latency**.

### 25.2.1. Memory Type

This options defines the structure of the on-chip memory:

- RAM (writable)—This setting creates a readable and writable memory.
- ROM (read only)—This setting creates a read-only memory.
- Dual-port access—This setting creates a memory component with two agents, which allows two hosts to access the memory simultaneously.

  *Note:* The memory component operates under true dual-port mode where both agent ports have address ports for read or write operations. If two hosts access the same address simultaneously in a dual-port memory undefined results will occur. Concurrent accesses are only a problem for two writes. A read and write to the same location will read out the old data and store the new data.

Intel Corporation. All rights reserved. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.
*Other names and brands may be claimed as the property of others.

ISO 9001:2015 Registered

- Single clock operation—Single clock operation setting creates single clock source to clock both agents port. If single clock operation is not selected, each of the two agents port is clocked by different clock sources.

  *Note:* For Intel Stratix 10 devices, only single clock operation is supported.

- Read During Write Mode—This setting determines what the output data of the memory should be when a simultaneous read and write to the same memory location occurs.
- Block type—This setting directs the Intel Quartus Prime software to use a specific type of memory block when fitting the on-chip memory in the FPGA.

  *Note:* The MRAM blocks do not allow the contents to be initialized during power up. The M512s memory type does not support dual-port mode where both ports support both reads and writes.

Because of the constraints on some memory types, it is frequently best to use the **Auto** setting. **Auto** allows the Intel Quartus Prime software to choose a type and the other settings direct the Intel Quartus Prime software to select a particular type.

## 25.2.2. Size

This options defines the size and width of the memory.

- Enable different width for Dual-port Access—Different width for dual-port access status.

  *Note:* A different width for dual-port access is not supported for Intel Stratix 10 devices.

- Agent S1 Data width—This setting determines the data width of the memory. The available choices are 8, 16, 32, 64, 128, 256, 512, or 1024 bits. Assign Data width to match the width of the host that accesses this memory the most frequently or has the most critical throughput requirements. For example, if you are connecting the on-chip memory to the data host of a Nios II processor, you should set the data width of the on-chip memory to 32 bits, the same as the data-width of the Nios II data host. Otherwise, the access latency could be longer than one cycle because the Avalon interconnect fabric performs width translation.
- Total memory size—This setting determines the total size of the on-chip memory block. The total memory size must be less than the available memory in the target FPGA.

  The IP parameter editor accepts characters **k** and **m** to specify the memory size in kilobytes and megabytes respectively. For example: if you enter 1k, it will automatically resolve to its equivalent bytes which is 1024 bytes in this case.

- Minimize memory block usage (may impact fmax)—Minimize memory block usage (may impact fmax)—This option is only available for devices that include M4K memory blocks. If selected, the Intel Quartus Prime software divides the memory by depth rather than width, so that fewer memory blocks are used. This change may decrease fmax.

## 25.2.3. Read Latency

On-chip memory components use synchronous, pipelined Avalon-MM agents. Pipelined access improves fMAX performance, but also adds latency cycles when reading the memory. The Read latency option allows you to specify either one or two cycles of read latency required to access data. If the Dual-port access setting is turned on, you can specify a different read latency for each agent. When you have dual-port memory in your system you can specify different clock frequencies for the ports. You specify this on the System Contents tab in Platform Designer.

### 25.2.4. ROM/RAM Memory Protection

This setting if enabled, creates additional reset request port for memory protection during reset. This additional reset input port is used to gate off the clock to the memory.

### 25.2.5. ECC Parameter

This setting if enabled, extends the data width to support ECC bits. It does not instantiate any ECC encoder or decoder logic within this component.

### 25.2.6. Memory Initialization

The memory initialization parameter section contains the following options:

- Initialize memory content—Option for user to enable memory content initialization.
- Enable non-default initialization file—You can specify your own initialization file by selecting Enable non-default initialization file. This option allows the file you specify to be used to initialize the memory in place of the default initialization file created by Platform Designer.
- Enable Partial Reconfiguration Initialization Mode—This setting if enabled, automatically instantiates logic to support Partial Reconfiguration use cases for initialized memory.
- Enable In-System Memory Content Editor Feature—Enables a JTAG interface used to read and write to the RAM while it is operating. You can use this interface to update or read the contents of the memory from your host PC.

## 25.3. Platform Designer System-Level Design for On-Chip Memory

There are few Platform Designer system-level design considerations for on-chip memories. See "Platform Designer System-Level Design".

When generating a new system, Platform Designer creates a blank initialization file in the Intel Quartus Prime project directory for each on-chip memory that can power up with initialized contents. The name of this file is `<name of memory component>.hex`.

## 25.4. Simulation for On-Chip Memory

At system generation time, Platform Designer generates a simulation model for the on-chip memory. This model is embedded inside the Platform Designer system, and there are no user-configurable options for the simulation testbench.

You can provide memory initialization contents for simulation in the file `<Intel Quartus Prime project directory>/<Platform Designer system name>_sim/<Memory component name>`**.hex**.

## 25.5. Intel Quartus Prime Project-Level Design for On-Chip Memory

The on-chip memory is embedded inside the Platform Designer system, and there are no signals to connect to the Intel Quartus Prime project.

To provide memory initialization contents, you must fill in the file `<name of memory component>`**.hex**. The Intel Quartus Prime software recognizes this file during design compilation and incorporates the contents into the configuration files for the FPGA.

*Note:* For the memory to be initialized, you then must compile the hardware in the Intel Quartus Prime software for the SRAM Object File (**.sof**) to pick up the memory initialization files. All memory types with the exception of MRAMs support this feature.

## 25.6. Board-Level Design for On-Chip Memory

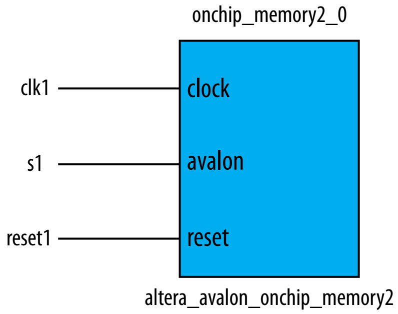
The on-chip memory is embedded inside the Platform Designer system, and there is nothing to connect at the board level.

## 25.7. Example Design with On-Chip Memory

This section demonstrates adding a 4 KByte on-chip RAM to the example design. This memory uses a single agent interface with a read latency of one cycle.

For demonstration purposes, the figure below shows the result of generating the Platform Designer system at this stage. In a normal design flow, you generate the system only after adding all system components.

**Figure 77. Platform Designer System with On-Chip Memory**



Because the on-chip memory is contained entirely within the Platform Designer system, Platform Designer `memory_system` has no I/O signals associated with `onchip_ram`. Therefore, you do not need to make any Intel Quartus Prime project connections or assignments for the on-chip RAM, and there are no board-level considerations.

## 25.8. On-Chip Memory (RAM and ROM) Core Revision History

| Document Version | Intel Quartus Prime Version | Changes |
|---|---|---|
| 2018.05.07 | 18.0 | Implemented editorial enhancements. |