

Summary - File Write

Name	file_write
Worker Type	Application
Version	v1.5
Release Date	4/2019
Component Library	ocpi.core
Workers	file_write.rcc file_write.hdl
Tested Platforms	isim, xsim, modelsim, xilinx13_3, xilinx13_4, centos6, centos7

Revision History

Revision	Description of Change	Date
v1.4	Initial release.	9/2019
v1.5	Changed all readable properties to readback. Also updated the end of file descriptions with smart wrapper updates.	4/2019

Functionality

The File_Write component writes application data to a file. It is normally used by specifying an instance of the File_Write component, and connecting its input port to an output port of the component produces the data. The name of the file to be written is specified in a property. This component has one input port whose name is “in”, which carries the messages to be written to the file. There is no protocol associated with the port, enabling it to be agnostic as to the protocol of the component connected to the input port.

Operating Modes

Data Streaming Mode

In data streaming mode, the contents of the file becomes the payloads of the stream of messages arriving at the input port. No message lengths or opcodes are recorded in the output file.

Messaging Mode

In messaging mode, the contents of the output file are written as a sequence of defined messages, with an 8-byte header in the file itself preceding the data for each message written to the file. This header contains the length and opcode of the message, with the data contents of the message following the header. The length can be zero, meaning that a header will be written but no data will follow the header in the file.

The first 32-bit word of the header is interpreted as the message length in bytes, little-endian. The next 8-bit byte is the opcode of the message, followed by 3 padding bytes. If the end of the file is encountered while reading a message header, or while reading the header-specified length of the message payload, an error will be reported and the component will terminate. The following Messaging File Field Layout shows the layout of a file:

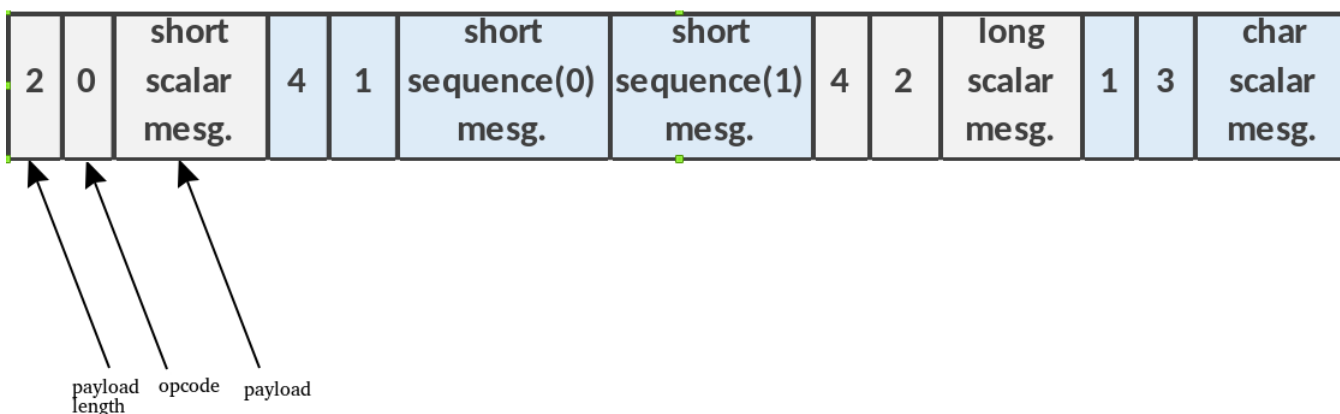


Figure 1: Messaging File Field Layout

No Protocol

The port on the component has no protocol specified in order to support interfacing with any protocol. This means that the created data file is formatted to match the protocol of the output port of the connected worker.

End of File Handling

When the File_Write component receives an end of file notification, it will interpret it as the end of data and declare itself “done”, and not write any further messages to the file.

Worker Implementation Details

file_write.hdl

This worker will only run on simulator platforms. This includes isim, xsim, and modelsim and will not run on or be built for any other hardware platforms. This is because it contains code that cannot be realized into RTL.

file_write.rcc

This worker is implemented in the C language version of the RCC model. Most new workers are implemented in the C++ language version of the RCC model.

Source Dependencies

file_write.rcc

- file_write.c
- file_write.h

file_write.hdl

- file_write.vhd

Component Spec Properties

Name	Type	SequenceLength	ArrayDimensions	Accessibility	Valid Range	Default	Usage
fileName	string length:1024	-	-	Initial	-	-	The name of the file that is written to disk from the input port.
messagesInFile	bool	-	-	Initial	-	false	The flag that is used to turn on and off message mode. See section on Message Mode.
bytesWritten	uLongLong	-	-	Volatile	-	-	The number of bytes written to file. Useful for debugging data flow issues.
messagesWritten	uLongLong	-	-	Volatile	-	-	The number of messages written to file. Useful for debugging data flow issues.
stopOnEOF	bool	-	-	Initial	-	true	Property has no functionality, exists for backwards compatibility purposes.

Worker Properties

file_write.hdl

Type	Name	Type	SequenceLength	ArrayDimensions	Accessibility	Valid Range	Default	Usage
Spec Property	fileName	string	-	-	Readback	-	-	added Readback
Property	CWD_MAX_LENGTH	ulong	-	-	Parameter	-	256	parameter for max string length for the cwd property
Property	cwd	string length:CWD_MAX_LENGTH	-	-	Volatile	-	-	the current working directory of the application (this is required for the hdl version of this worker and cannot be determined automatically)

file_write.rcc

No properties changed from component specification.

Component Ports

Name	Producer	Protocol	Optional	Advanced	Usage
in	False	None	False	-	-

Worker Interfaces

file_write.hdl

Type	Name	DataWidth	Advanced	Usage
StreamInterface	in	32	-	-

file_write.rcc

Type	Name	DataWidth	Advanced	Usage
Port	in	-	bufferSize=8k	-

Test and Verification

All test benches use this worker as part of the verification process. A unit-test does not exist for this component.