AMD
XILINX

Vivado Design Suite User Guide : Designing with IP (UG896)

Using a COE File

COE File Syntax
COE File Examples
MIF File Description

Using a COE File

In certain cases, some parameter values are passed to the Vivado IP catalog using a COE (COEfficient) file; an ASCII text file with a single radix header followed by several vectors. The radix can be 2, 10, or 16. Each vector must be terminated by a semi-colon.
The Vivado tool reads the COE file and writes out one or more MIF files when the core is generated. The VHDL and Verilog behavioral simulation models for the core rely on these MIF files.

!! Important: You must upgrade all IP prior to adding a COE file. Additionally, locate the COE file in the same directory as the XCI file.

Note: If a COE file is no longer used by an IP, remove the file. Failure to remove an old COE file can result in both the newly associated COE and the old COE being passed to synthesis. Additionally, if the old COE is removed from disk, but not from the project, an error occurs during synthesis.

COE File Syntax

The following syntax displays the general form for a COE file:

```
Keyword =Value ; Optional Comment
Keyword =Value ; Optional Comment
<Radix_Keyword> =Value ; Optional Comment
<Data_Keyword> =Data_Value1, Data_Value2, Data_Value3;
```

The following table describes COE file keywords for specifying radix values for data. Keywords are not case-sensitive. For information on the specific keywords required for a IP, see the Product Guide for that IP.

Table: COE File Keywords for Radix Values

Keyword	Description
`RADIX`	Used for non-memory cores to indicate the radix being used to specify the coefficients of the filter.

Keyword	Description
MEMORY_INITIALIZATION_RADIX	Used for memory initialization values to specify the radix used.

The following table describes COE file keywords for data values. Keywords are not case sensitive.

Table: COE File Keywords for Data Values

Keyword	Description
COEFDATA	Used for filters to indicate that the data that follows comprises the coefficients of the filter.
MEMORY_INITIALIZATION_VECTOR	Used for block and distributed memories.
PATTERN	Used for Bit Correlator COE files.
BRANCH_LENGTH_VECTOR	Used in Interleaver COE files.

Note: Any text after a semicolon is treated as a comment and ignored.

One of the following keywords must be the last keyword specified in the COE file:

- COEFDATA
- MEMORY_INITIALIZATION_VECTOR

Any other keywords that follow are ignored.

COE File Examples

Virtex Bit Correlator COE File Example

```
************************************************************
************ Example of Virtex Bit Correlator.COE ************
************************************************************
; Sample .COE coefficient file for v2.0 and later
; versions of the Bit Correlator core.
;
; In this core, a COE file is used to specify the value
```

```
; of the bit mask when the Pattern Mask option is selected.
;
; Specifications:
;
; - 19 taps, hexadecimal coefficients
; - Serial input data
;
; Please refer to the datasheet for this core for more
; details on using the Mask option.
radix = 16;
pattern = 3 0 3 1 0 1 1 3 0 2 2 2 3 0 1 1 3 0 3;
```

Dual Port Block Memory COE File Example

```
********************************************************************
********* Example of Dual Port Block Memory .COE file **********
********************************************************************
; Sample memory initialization file for Dual Port Block Memory,
; v3.0 or later.
;
; This .COE file specifies the contents for a block memory
; of depth=16, and width=4. In this case, values are specified
; in hexadecimal format.
memory_initialization_radix=2;
memory_initialization_vector=
1111,
1111,
1111,
1111,
1111,
0000,
0101,
0011,
0000,
1111,
1111,
1111,
1111,
1111,
```

```
1111,
1111;
```

Single Port Block Memory .COE file Example

```
**************************************************************
******** Example of Single Port Block Memory .COE file *********
**************************************************************
; Sample memory initialization file for Single Port Block Memory,
; v3.0 or later.
;
; This .COE file specifies initialization values for a block
; memory of depth=16, and width=8. In this case, values are
; specified in hexadecimal format.
memory_initialization_radix=16;
memory_initialization_vector=
ff,
ab,
f0,
11,
11,
00,
01,
aa,
bb,
cc,
dd,
ef,
ee,
ff,
00,
ff;
```

Distributed Memory .COE File Example

```
**************************************************************
************ Example of Distributed Memory .COE file ***********
**************************************************************
; Sample memory initialization file for Distributed Memory v2.0 and
```

```
; later.
;
; This .COE file is NOT compatible with v1.0 of Distributed Memory Core.
;
; The example specifies initialization values for a memory of depth= 32,
; and width=16. In this case, values are specified in hexadecimal
; format.
memory_initialization_radix = 16;
memory_initialization_vector = 23f4 0721 11ff ABe1 0001 1 0A 0
 23f4 0721 11ff ABe1 0001 1 0A 0
 23f4 721 11ff ABe1 0001 1 A 0
 23f4 721 11ff ABe1 0001 1 A 0;
*********************************************************************
****** Example of Distributed Arithmetic FIR Filter .COE file ***
*********************************************************************
; Example of a Distributed Arithmetic (DA) FIR Filter .COE file
; with hex coefficients, 8 symmetrical taps, and 12-bit
; coefficients.
;
; Compatible with all versions of the Distributed Arithmetic
; FIR Filter which supports Virtex and Spartan
Radix = 16;
CoefData= 346, EDA, 0D6, F91, F91, 0D6, EDA, 346;
```

MIF File Description

The COE file provides a high-level method for specifying initial memory contents. When the core is generated the Vivado tools convert the COE file into a MIF file, which holds the actual binary data used to initialize the memory in the core and simulation models.

The MIF file consists of one line of text per memory location. The first line in the file corresponds to address 0, and the second line corresponds to address 1, and so forth. The text on each line must be the initialization value (MSB first) for the corresponding memory address in binary format, with exactly one binary digit per bit of memory width.

✎ **Note:** For HDL simulations, the MIF file must reside in the simulation directory.