In this paper, we aim to detect pragma-related bugs in Vitis HLS. We use PICT to generate pragma combinations to test Vitis HLS. However, some pragma combinations cannot pass the check mechanism in Vitis HLS. For example, in some cases, applying ***array_reshape*** and ***array_partition*** to an array at the same time may also cause 'pre-synthesis failed' in Vitis HLS. The usage of ***array_reshape*** and ***array_partition*** is also not recommended by developers. Therefore, some conditions are set to limit PICT to generating pragma combinations which can be processed by HLS tools normally. The format of conditions is as follows:

IF [***component a***] = " ***xxx***" THEN [***component b***] <> " ***yyy***";

The format means when pragma ***xxx*** is selected for ***component a***, pragma ***yyy*** will never be selected for ***component b***. For example, when we write statement (1) into the input file of PICT as shown in Fig.7 (b), PICT will not generate pragma combinations with ***array_reshape*** and ***array_partition*** selected for array ***l_135*** simultaneously.

IF [***l_135-1***] ="#pragma HLS ***array_reshape***" THEN [***l_135-2***] <> "#pragma HLS ***array_partition***"; (1)

In this paper, we summarize many conditions (***xxx*** and ***yyy***) which are shown below.

(1). When ***component a*** and ***b*** are related to the same array, ***xxx*** and ***yyy*** can be set as shown below.

xxx	***yyy***
#pragma HLS array_reshape	#pragma HLS array_partition
#pragma HLS array_reshape	#pragma HLS stream
#pragma HLS array_partition	#pragma HLS stream

(2). When ***component a*** and ***b*** are related to the same loop, ***xxx*** and ***yyy*** can be set as shown below.

xxx	***yyy***
#pragma HLS dataflow	#pragma HLS pipeline
#pragma HLS dataflow	#pragma HLS unroll
#pragma HLS dataflow	#pragma HLS loop_flatten
#pragma HLS dataflow	#pragma HLS latency
#pragma HLS latency	#pragma HLS loop_flatten
#pragma HLS latency	#pragma HLS pipeline
#pragma HLS protocol	#pragma HLS loop_flatten

(3). When ***component a*** and ***b*** are related to the same function, ***xxx*** and ***yyy*** can be set as shown below.

xxx	***yyy***
#pragma HLS dataflow	#pragma HLS pipeline

(4). When ***component a*** is related to an array and ***component b*** is related to a function which is called behind ***a*** **in the same function**, ***xxx*** and ***yyy*** can be set as shown below.

xxx	***yyy***
#pragma HLS array_reshape	#pragma HLS inline
#pragma HLS array_reshape	#pragma HLS pipeline
#pragma HLS array_reshape	#pragma HLS expression_balance
#pragma HLS array_reshape	#pragma HLS loop_merge
#pragma HLS array_partition	#pragma HLS inline
#pragma HLS array_partition	#pragma HLS pipeline
#pragma HLS array_partition	#pragma HLS expression_balance
#pragma HLS array_partition	#pragma HLS loop_merge

(5). When ***component a*** is related to an array and ***component b*** is related to the if region where a is **in the same funtion**, ***xxx*** and ***yyy*** can be set as shown below.

xxx	***yyy***
#pragma HLS array_reshape	#pragma HLS protocol
#pragma HLS array_partition	#pragma HLS protocol
#pragma HLS array_reshape	#pragma HLS occurrence
#pragma HLS array_partition	#pragma HLS occurrence

(6). When ***component a*** is related to an array and ***component b*** is related to the loop region where a is **in the same function**, ***xxx*** and ***yyy*** can be set as shown below.

xxx	***yyy***
#pragma HLS array_reshape	#pragma HLS protocol
#pragma HLS array_partition	#pragma HLS protocol

(7). When ***component a*** is related to a function and ***component b*** is related to a function which is called by ***a***, ***xxx*** and ***yyy*** can be set as shown below.

xxx	***yyy***
#pragma HLS dataflow	#pragma HLS pipeline

(8). When ***component a*** is related to a function and ***component b*** is related to a loop in ***a***, ***xxx*** and ***yyy*** can be set as shown below.

xxx	***yyy***
#pragma HLS dataflow	#pragma HLS pipeline
#pragma HLS dataflow	#pragma HLS dataflow

(9). When ***component a*** is related to a loop and ***component b*** is related to the loop in ***a***, ***xxx*** and ***yyy*** can be set as shown below.

xxx	***yyy***
#pragma HLS dataflow	#pragma HLS pipeline
#pragma HLS dataflow	#pragma HLS dataflow
#pragma HLS dataflow	#pragma HLS latency
#pragma HLS dataflow	#pragma HLS protocol
#pragma HLS dataflow	#pragma HLS unroll

Although we use some restrictions, we do not think it will affect the results of the experiment because TEPACS and its variants use the same restrictions in the experiment. The restrictions we summarized are also helpful for other researchers to look deeper for pragma-related bugs in HLS tools.

Note that, the format we use is specific to PICT, researchers should write condition statements based on the tools they use.