Lab3: Adaptive Threshold Engine

1. Introduction

Threshold is the simplest method of image segmentation. From a grayscale image, threshold can be used to create binary images

Please design an Adaptive Threshold Engine (ATE). As shown in Fig 1, the function of the ATE circuit is used to separate a grayscale image from the foreground image. The system block diagram is shown in Fig 2. Its detailed specifications will be described later. Each input and output signal refers to **Table 1**.

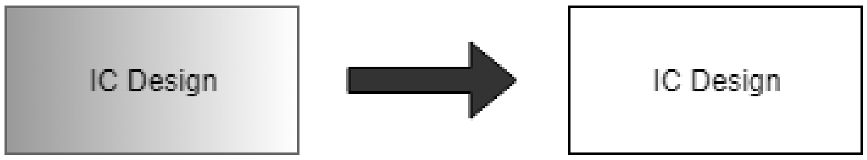


Fig 1. The function of Adaptive Threshold Engine

2. Design Specifications

Block Overview

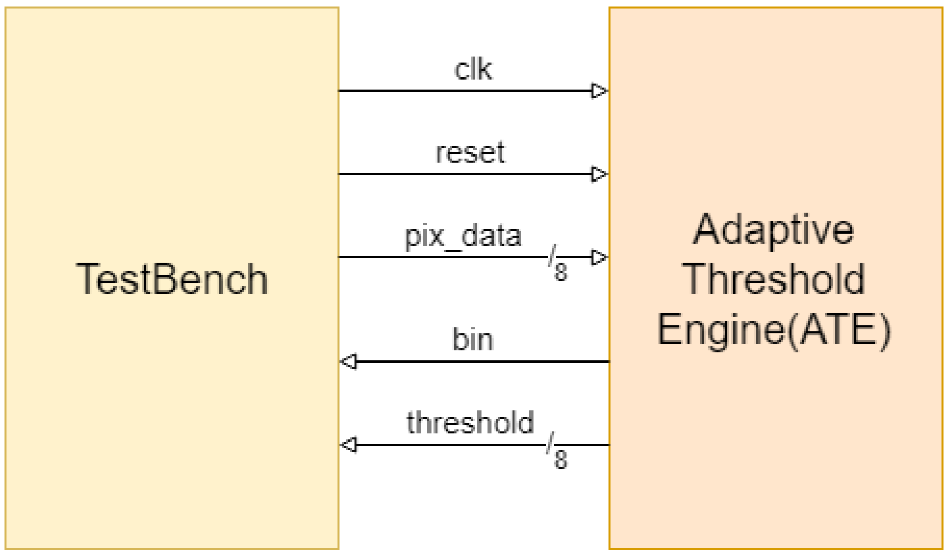


Fig 2. Block Overview

I/O Interface

Table 1. I/O Interface

Signal Name	I/O	Width	Description
clk	I	1	clock for the computational system
reset	I	1	reset the state of the computational system when it asserts
pix_data	I	8	the 8-bit input pixel data
bin	O	1	each pixel data is converted to 1-bit data by threshold
threshold	O	8	threshold data for single region

System Description

Image input

The size of input image is 48X32 (6X4 block). Each block has 8X8 points. The image input order is sequentially, as shown in the block number sequence of **Fig 3.**

When performing Threshold processing, it takes the block-base as a unit. The leftmost and rightmost lines do not need to be processed, so their bin and threshold are all outputs 0. As shown in Fig 3, blocks 0, 6, 12, 18 and The 5, 11, 17, 23 blocks do not need to be processed.

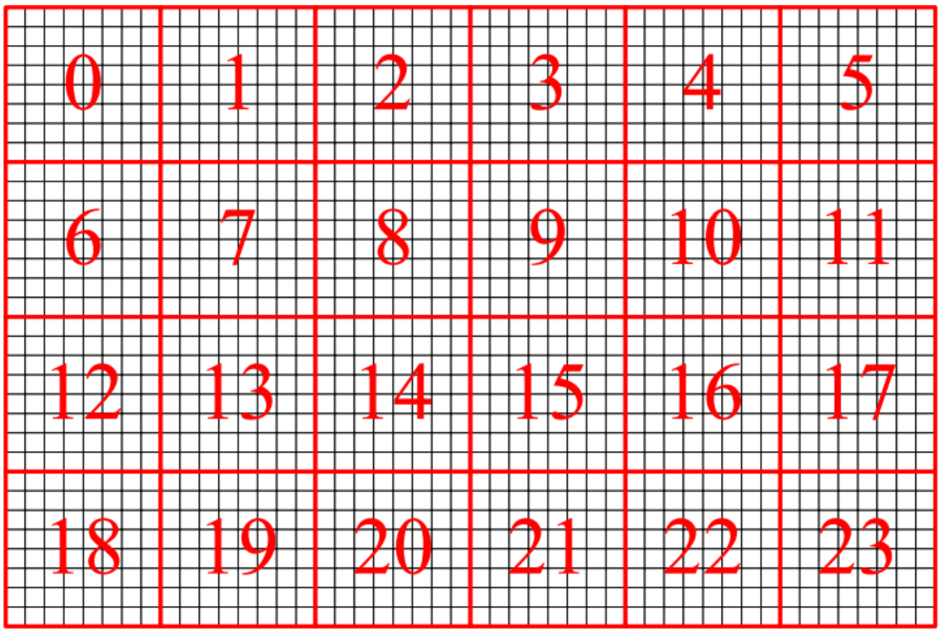


Fig 3. Input image block

Adaptive Threshold

The purpose of threshold is to distinguish the foreground and background of a grayscale image. For each point of the image block, if it is greater than or equal to the threshold value of the block, it outputs 1 and vice versa. The Adaptive threshold indicates that this threshold is calculated.

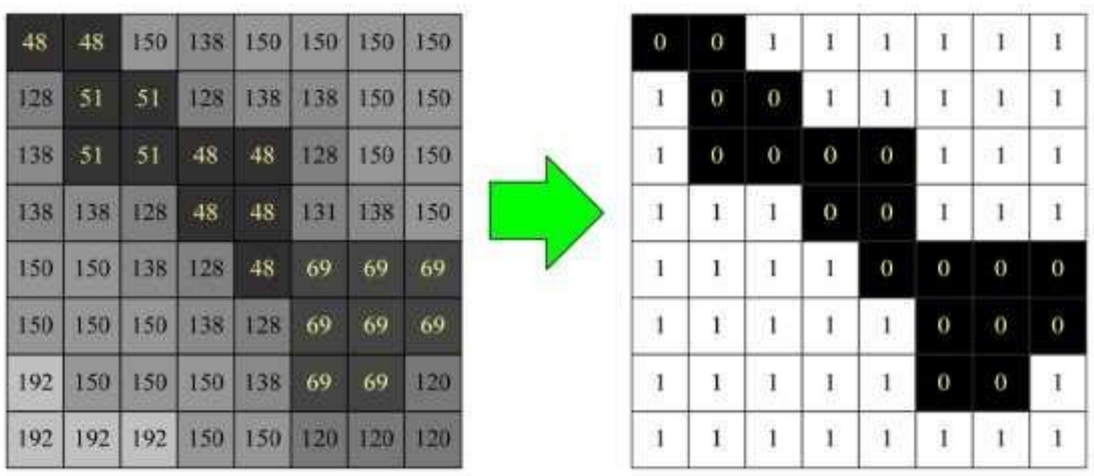
The threshold calculation method for this ATE circuit uses the average of the maximum and minimum values in a single 8x8 block, i.e. threshold=(Max+Min)/2, and if the threshold is a floating point number, **the unconditional carry is taken**.

For each point:

bin=1 if pix_data >= threshold

bin=0 if pix_data < threshold

In **Fig 4**, for example, the maximum value of the block is 192, and the minimum value is 48. Then the threshold = (192+48)/2=120, so the output is the right picture.



48	48	150	138	150	150	150	150
128	51	51	128	138	138	150	150
138	51	51	48	48	128	150	150
138	138	128	48	48	131	138	150
150	150	138	128	48	69	69	69
150	150	150	138	128	69	69	69
192	150	150	150	138	69	69	120
192	192	192	150	150	120	120	120

0	0	1	1	1	1	1	1
1	0	0	1	1	1	1	1
1	0	0	0	0	1	1	1
1	1	1	0	0	1	1	1
1	1	1	1	0	0	0	0
1	1	1	1	1	0	0	0
1	1	1	1	1	0	0	1
1	1	1	1	1	1	1	1

Fig 4. Single block threshold

Timing Diagram

Output Order

The test circuit will be reset at the beginning. After the reset is completed, the entire image content will be input in an uninterrupted manner. The order of the input blocks is from left to right, from top to bottom. In a single block, the value of each point is also from left to right and from top to bottom. The order of output is the same as the order of input, but the output will add an output delay time one block later than the input, and the output will be uninterrupted.

As shown in **Fig 5**, the output delay is set to 1 cycle. When the test circuit inputs the content of the first block, the ATE sends the calculation result of the first block at the next cycle.

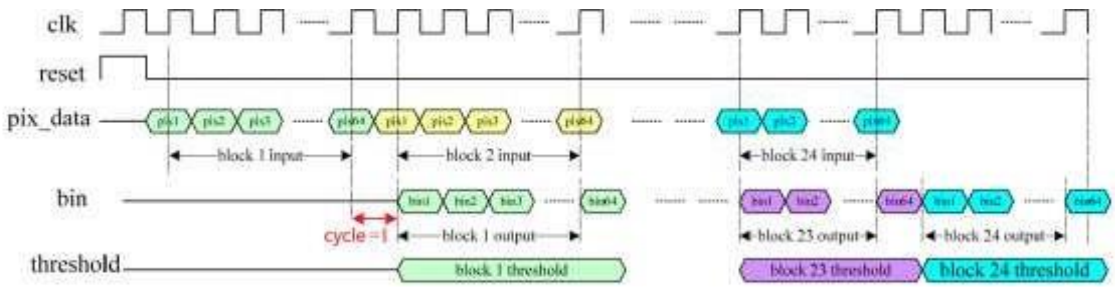
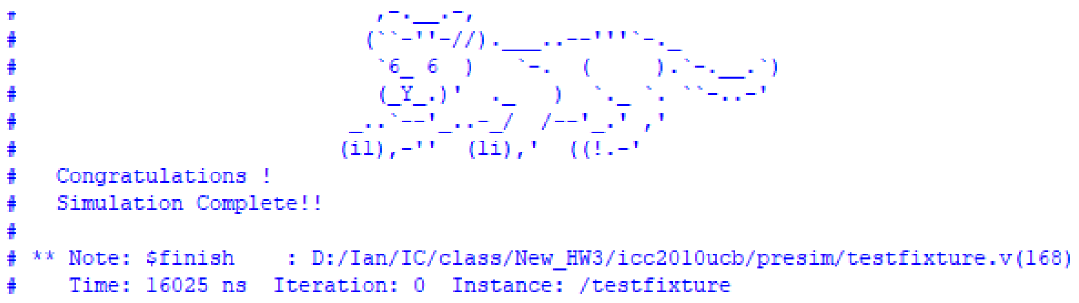


Fig 5. Output order

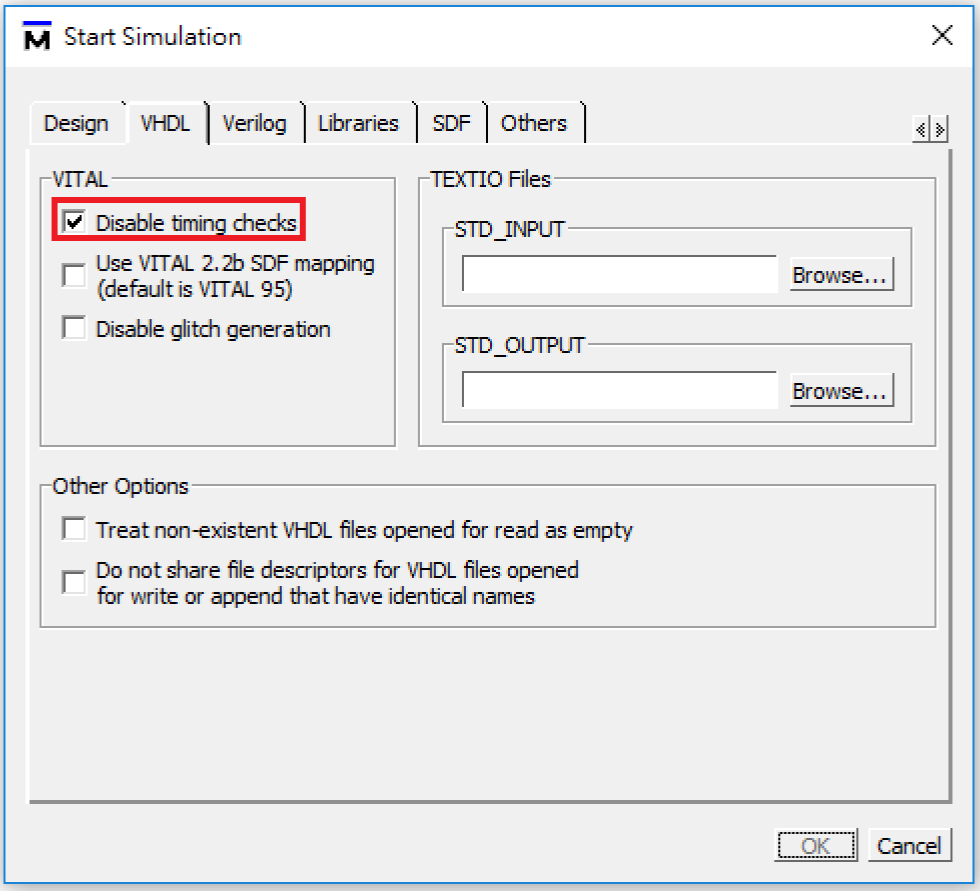
3. Simulation

Functional Simulation

All of the result should be generated correctly, and you will get the following message in ModelSim simulation. You can turn off the timing check in pre-sim only.



```
#                            ,-.__.-,
#                           (``-'''-//).___..--'''`-._
#                            `6_ 6  )   `-.  (     ).`-.__.`)
#                            (_Y_.)'  ._   )  `._ `. ``-..-'
#                          _..`--'_..-_/  /--'_.' ,'
#                         (il),-''  (li),'  ((!.-'
#   Congratulations !
#   Simulation Complete!!
#
# ** Note: $finish    : D:/Ian/IC/class/New_HW3/icc2010ucb/presim/testfixture.v(168)
#    Time: 16025 ns  Iteration: 0  Instance: /testfixture
```



Start Simulation

Design | VHDL | Verilog | Libraries | SDF | Others

VITAL
- [✓] Disable timing checks
- [ ] Use VITAL 2.2b SDF mapping (default is VITAL 95)
- [ ] Disable glitch generation

TEXTIO Files
- STD_INPUT: Browse...
- STD_OUTPUT: Browse...

Other Options
- [ ] Treat non-existent VHDL files opened for read as empty
- [ ] Do not share file descriptors for VHDL files opened for write or append that have identical names

OK | Cancel