

ECE 545

Midterm Exam 2

Fall 2005

Problem

Function

Design and describe using RTL VHDL a circuit capable of processing two streams of up to 2^m-1 data bytes in each, in search for the longest matching subsequence, and its position within the data streams. Whenever there is a match between an ASCII character at the A_data input, and the ASCII character at the B_data input, the match_nr output is incremented. Matching between two characters can be done in the case sensitive way or case insensitive way, depending on the value of the control input case_insensitive. The output max_match_nr contains the maximum value at the output of match_nr since the last reset. The output max_match_pos contains the position of the beginning of the longest matching sequence.

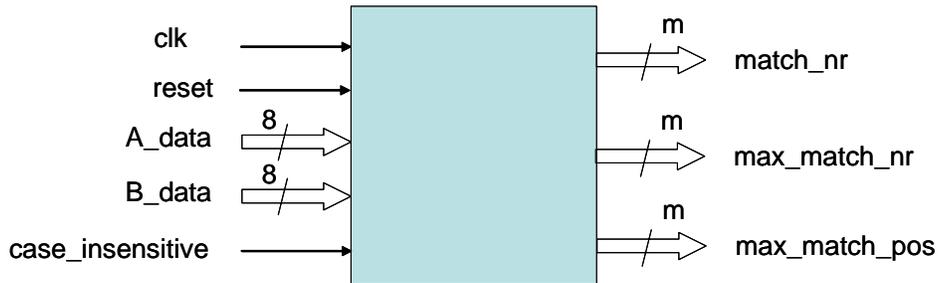
Example of the circuit operation is given below:

```

A_data      AdfdYyHdad;-454tyll;fd;lghhfg;lsgm
B_data      ydfduYhgghd;-488hgfoKFD;LghhgGoffgm
case_insensitive 00000000000000001111111111111111
match_nr    0123000001234000000012345670100012
max_match_nr 012333333333444444444444445677777777
  
```

Interface

Assume the following interface to your circuit:



| Port | Width | Meaning |
|------------------|-------|--|
| clk | 1 | System clock |
| reset | 1 | System reset – clears all internal registers and counters |
| A_data | 8 | Input for the data stream A |
| B_data | 8 | Input for the data stream B |
| case_insensitive | 1 | Value '1' means that comparison is case insensitive, '0' means that it is case sensitive |
| match_nr | 1 | Number of consecutive positions in both data streams that match each other |
| max_match_nr | 1 | Maximum number of consecutive positions in both data streams that have match each other |
| max_match_pos | 1 | Position of the beginning of the longest matching sequence |

Optimization

Optimize your circuit for the maximum data throughput. When choosing between two circuits with the same or very similar throughput, give preference to the circuit with the smaller area.

Verification

Verify your circuit using functional simulation for the case of $m=8$ using a testbench capable of reading input data from a text file in the hexadecimal notation, and writing output data to a text file in the hexadecimal notation.

Synthesis

Synthesize your circuit using Synopsys for the two cases: $m=8$ and $m=16$, using the 90 nm TCBN90G TSMC library

Based on the synthesis reports, determine for each case:

- a. circuit area
- b. minimum clock period
- c. maximum input data throughput

Suggested design steps

1. Draw a block diagram of the execution unit of your circuit
2. Draw a block diagram of a part of the control unit of your circuit
3. Describe the remaining portion of the control unit using ASM chart
4. Translate ASM chart from actions to values of specific control signals
5. Describe an interface to your circuit in VHDL
6. Translate block diagrams designed in steps 1 and 2 to VHDL
7. Translate the final ASM chart obtained in step 4 to VHDL
8. Write a testbench capable of simulating your circuit
9. Verify the operation of your circuit using functional simulation, introduce corrections in your code if necessary
10. Synthesize and implement your circuit.
11. Determine circuit area, minimum clock period, and worst case execution time based on the implementation reports.

Deliverables:

Handwritten on paper, scanned, and converted into a JPG file, or in the form of a Power Point, MS Word, or a PDF file

1. All block diagrams you have developed
2. All ASM charts you have developed

Electronic version

1. All source codes describing your circuit
2. All testbench files, including input and corresponding output files
3. All Synopsys script files and post-synthesis reports
4. A short text or MS Word file describing circuit area, minimum clock period, and maximum data throughput.
5. Waveforms from the functional simulation.