COMP3211/9211 Computer Architecture Assignment

Application Specific Processor for Real-Time Pattern Search

This assignment is a design project to be completed by your lab group. It is due in Week 10 (see detailed information provided in this document).

Project Description

Pattern searching is a typical problem in many application areas. This project is to design a processor system for searching patterns in an input string stream and output the number of occurrences of each pattern that has been found so far, as outlined in Figure 1 (a). An example of searching one pattern "*ab*" is illustrated in Figure 1(b). Each time a matched pattern is found, the occurrence count *N* is incremented by 1. Note: the time that is required for finding a matched pattern is not shown in the illustration.

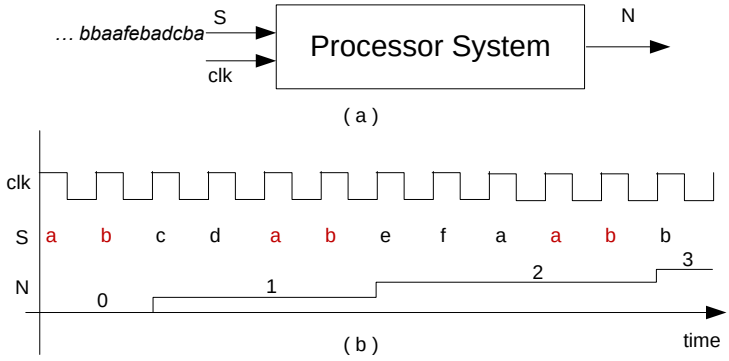

Figure 1

For this project, we have the following assumptions:

- A pattern is made of alphabetical letters and numerical digits. The pattern size can be up to 16 characters. The pattern is not case sensitive and symbol '?' can be used in the pattern to present any letter. For example, a?8b represent any 4-character patterns that starts with letter 'a' and ends with "8b". The processor can search up to 8 patterns in the input string concurrently.

- Overlapped patterns are counted separately. For example, string "ababacef" is regarded to contain two "aba" patterns.

- Both input string and patterns are initially stored in two separate files. To simulate the feature of real-time processing, characters in the input string file are sequentially fetched/read by the testbench and stored in a special I/O register that can be accessed by the processor.

- The I/O register is small and can only hold 4 characters. To prevent overrun error (where characters saved to the I/O register are overwritten by the new coming characters before they are processed by the processor), the speed of input stream should match the processing speed of the processor.

- The I/O register can also be used to load all patterns into memory before the real-time string search starts. The size of each pattern should be determined by the processor based on the pattern provided.

- During execution, the length of the input string processed so far and the occurrence count of each pattern are kept in a table. The table is stored in the memory and updated in real time with the incoming stream.

- The memory used in the system is small and has the same speed of the processor.

For information that is not provided here but you think you need to consider in your design, you can make reasonable assumptions.

Detailed Requirements

For this assignment, you need to

- **design an instruction set** that is easy to implement and effective so that the processor can be programed for the functions specified above,

- **write machine code** for the real-time pattern search,

- **build a pipelined processor model** for your instruction set (in VHDL or Verilog),

- **create a test testbench, input data files and testing strategies** to test your design for varied cases, for example,
    - patterns with wildcard, multiple patterns of different sizes,
    - input string containing overlapped patterns,
    - cases that can demonstrate the unique features of your design.

- **simulate and verify your design**, and

- **analyze the performance and cost of your design**.

Guide to Project Development

Each project group is required to make a detailed plan that may include:

- tasks that need to be performed in order to complete this assignment,

- a schedule of the tasks,

- the role of each member for the tasks,
- a test method for each task,
- a project management strategy to ensure that the design project is carried out smoothly and completed on time with a quality as good as possible.

Assessments (total 100 marks):

The assessment for each group consists of

- group presentation (30%)
  - assessed by the peers in your TLB class in Week 10. Each group will be given 25 mins.
  - A form will be available for the peer assessment.
- lab demonstration (50%)
  - assessed by tutor in your TLB class in Week 10
- report (20%)
  - *Give* submission of code and report due on Monday, Week 11 (April 24)
    - zip your Vivado project and the project report. **Each group only needs to submit one copy**. The submission system will be set up on the Assignment page of the course website in Week 10.
  - Note: Report is a written form of the project design. It should be extensive in content, effective in use of figures and tables, and concise and succinct in description.

Each member needs to submit a form about the contribution of all the members (including themselves) in the group **before Friday Week 11**. On submission of this form, make sure the total contribution of all the members adds up to 100%.

The calculation of your individual marks of the assignment is based on two components: the mark of whole group (G) and your relative contribution (C). It is calculated as *G*C* but capped by 100, as demonstrated in the following two examples:

- For a group of four members, if everyone contributes equally and the group has a mark is 90, each member's mark will be *max{90*(100%/4)/(100%/4), 100}=90*.
- For a group of four members with a group mark of 70, if individual contributions are 10%, 20%, 20%, and 50%, respectively, the one with the highest contribution will receive a mark of *max{70*50%/(100%/4), 100}=100*.