

1. [20 points] Read the paper “Hardware/Software Partitioning of Software Binaries” by G. Stitt and F. Vahid published in 2002 IEEE/ACM international conference on Computer-aided design. Answer the following questions in your own words:

- a) [5 points] What is the main idea of this paper and how is it different than techniques discussed in the lecture?
- b) [10 points] What are the advantages and disadvantages of partitioning from software binaries compared to program source code?
- c) [5 points] Explain how the authors interfaced their reconfigurable circuits with the main processor.

2. [30 points] One of the popular functions to accelerate using reconfigurable computing is the *Edit Distance* function. The edit distance function measures the *difference* between two strings by *aligning* them and estimating the cost of transforming one string, character by character, to the other. Consider two strings: TTGACATC and TTCAATGC. When comparing the two strings (please see the next figure), then at each position one of three cases can occur:

```

T T G A C A T   C
| |   |   | |   |
T T C A   A T G C

```

- a. a *match* can occur when the same character is present in both strings.
- b. a *mismatch* when there are two different characters at the same position in the two strings.
- c. A *gap* where there is an *insertion* of one character in only one string, or symmetrically a *deletion* in the other string.

An effective way to solve such problem is through *dynamic programming* which breaks the given problem into smaller sub problems and then solves the subproblems and combines the results to obtain the final solution. If we have two strings $X = \{x_1, x_2, \dots, x_n\}$ and $Y = \{y_1, y_2, \dots, y_n\}$ then we can define the function $H(i, j)$ as the minimum difference between the two substrings $x_1x_2\dots x_i$ and $y_1y_2\dots y_j$. $H(i, j)$ can be calculated as follows.

$$H(i, j) = \min \begin{cases} H(i-1, j-1) + \text{cost of matching/mismatching } x_i \text{ to } y_j \text{ (} = c(x_i, y_j) \text{)} \\ H(i, j-1) + \text{cost of matching a gap ' _ ' in } X \text{ to } y_j \text{ (} = c(_, y_j) \text{)} \\ H(i-1, j) + \text{cost of matching a gap ' _ ' in } Y \text{ to } x_i \text{ (} = c(x_i, _) \text{)} \end{cases}$$

A straight-forward pseudo-code to calculate the difference between the two strings is as follows

```
initialize array H[0→n] [0→n] to zero;

for (i = 1; i <= n; i++)
begin
  for(j = 1; j <= n; j++)
  begin
    H[i][j] = min(H[i-1][j-1]+c(x[i], y[j]),
                  H[i][j-1]+c(_, y[j]),
                  H[i-1, j]+c(x[i],_));
  end
end
```

Assume the presence of an instruction (**min**) that takes five inputs $H[i-1][j-1]$, $H[i-1][j]$, $H[i][j-1]$, $x[i]$, $y[j]$ and produces the minimum result in 1 cycle.

- [15 points] Assuming $n = 3$, perform loop unrolling for both the outer and inner loops. Write down the sequence of **min** instructions that results from loop unrolling.
- [15 points] Construct the data flow graph for the **min** instructions sequence that results from part (b).

Side note: The genome of living organisms can be symbolically represented by a string of four letters (A, C, G, T). Given two strings that correspond to the genomes of two organisms, the edit distance becomes a powerful tool in determining the closeness or similarity of the two organisms.