# Solutions - Homework 3

(Due date: Oct. 31st)

## PROBLEM 1 (20 PTS)

- Refer to Activity 6 in the *High-Performance Embedded Programming with the Intel® Atom™ platform → Tutorial 5*
  - ✓ Activity 6 – Grayscale Image Morphology: Execute the application so that it generates the `uchip_d.bof` and `uchip_e.bof` files. Provide a screenshot of the execution in the Terminal (erosion or dilation) and complete Table I. (20 pts)
    ** Embed the image in your Homework 3 document.*

TABLE I. COMPUTATION TIME (US) OF DILATION/EROSION. DE2I-150 BOARD

| | Computation Time (us) | |
|---|---|---|
| | Sequential | TBB |
| Dilation | | |
| Erosion | | |

## PROBLEM 2 (30 PTS)

- In the following code snippet, we apply this transformation to the elements of the vector $\vec{x}$. The result is a vector $\vec{r}$:

$$r(i) = \frac{1}{1 + e^{-x(i)}}, i = 0, \ldots, n-1$$

```
...
double tmp;
double *x, *r;
x = (double *) calloc (1000,sizeof(double));
r = (double *) calloc (1000,sizeof(double));

tbb:parallel_for (int(0), int(1000), [&] int i) {
   tmp = 1 + exp(-x[i]);
   r[i] = 1/tmp;
   } );
...
```

✓ If your own words, explain why this code might not generate correct results all the time.

Race conditions

✓ How would you fix the code so that correct results are guaranteed?

Encapsulate everything in a function, or make everything dependent on i.

# PROBLEM 3 (30 PTS)

- Refer to the Activity 1 in the *High-Performance Embedded Programming with the Intel® Atom™ platform → Tutorial 7*
  - ✓ Activity 1 – Modulus. Execute the application. Provide a screenshot of the execution in the Terminal. (10 pts)
  ** Embed the image in your Homework 3 document.*

- Based on the completion of the Activity 1 (3-stage pipeline), answer the following questions: (20 pts)
  - ✓ Stage 1, whose functor is `my_in(a,b,n)`, has no input (only a `flow_control` object is passed to the functor). Stage 1 feeds input data items into the pipeline and notifies the pipeline when there are no more items in the input stream.
    - What type are the input data items? How many bytes does an input data item occupy?

      Type: MyMod. Each input data item occupies 8 bytes

    - Where in the functor definition code (you can copy and paste the code line) is the output of Stage 1 generated?
      if (i < n) { t.av = *(a+i); t.bv = *(b+i); i++; return t; }

  - ✓ Stage 2: Its associated functor is `my_transf()`. This functor has no input parameters.
    - Where in the functor definition code does Stage 2 read its incoming data?

      float operator() (MyMod input) const

    - Where in the functor definition code is the output of Stage 2 generated?

      return result;

  - ✓ Stage 3, whose functor is `my_out(c)`:
    - Is the input data to Stage 3 the same as the input parameters of the functor?

      No

    - Where in the functor definition code does Stage 3 read its incoming data?

      void operator () (float result) const {

    - Since, syntax-wise, Stage 3 has no output, how do we store the result for each incoming data item?
      A parameter c is passed. This is a pointer to where we can modify contents

# PROBLEM 4 (20 PTS)

- Attach your Project Status Report (no more than 1 page, single-spaced, 2 columns, only one submission per group). This report should contain the initial status of your project. For formatting, use the provided template (`Final Project – Report Template.docx`). The sections included in the template are the ones required in your Final Report. At this stage, you are only required to:
  - ✓ Include a title and a (draft) project description. (10 pts)
  - ✓ Your application should be explained in an algorithmic fashion (i.e., like pseudo code and/or flowchart) (5 pts)
  - ✓ Include the parallelization strategy you plan to apply in your application. Here, a draft idea suffices. You can always modify this as you keep working on your application. (5 pts)

- Only one student is needed to attach the report (make sure to indicate all the team members).