# Computer architecture

## Homework week 8 - BONUS

# Instructions

Submit by e-mail to the lecturer, as a PDF document with your name and student ID near the beginning. You can work on this in groups of 2 people. Use the English language. Deadline: Nov 4th, 23:59.

The grade of this assignment can be used to replace a grade of another assignment.

# Question 1 - Memory address ranges in MGSim (2pt)

1. Make a (simplified) diagram of the I/O interconnect implemented by MGSim.
2. Using the following commands in the MGSim interactive prompt (`-i`), draw a map of the memory address space:

```
info memory ranges
info cpu0.mmio
info cpu0.io_if.aio
info cpu0.io_if.pnc
info iobus0
```

> **Note**
>
> The "map" of an address space can be drawn either as a ribbon with segments colored/labeled based on their role; or as a table with address ranges in one column and descriptions in another.

# Question 2 - Graphical output (4pt)

Consider the following program, to be used with the `minisim.ini` and `minicrt` from practicumopdracht 1/2:

```
typedef unsigned int uint32_t;

volatile uint32_t *ctl = 0x46000000;
volatile uint32_t *fb = 0x47000000;
```

```
int main(void)
{
  int i;
  ctl[1] = 320;
  ctl[2] = 240;
  ctl[3] = 32;
  ctl[0] = 1;

  for (i = 0; i < 320*240; i += 319)
      fb[i] = 0xff0000;

  ctl[4] = 0;

  return 0;
}
```

Using either the Alpha MGSim or your own MIPS implementation, ensure you can compile, link and run this program. Check out the `.ppm` file produced in the same directory after it runs.

1. Review the document `mgsimdev-gfx` provided alongside the MGSim source code, in the `doc/` subdirectory. Then comment line by line what the program above does.
2. Modify this program to draw a triangle (any color, any position: you choose).
3. Extend your answer to #2 to generate 10 different PPM files with the triangle in different positions (ie an animation). Feel free to draw more interesting stuff if so desired.

# Question 4 - throughput (4pt)

The following code can be used to benchmark a section of your program from Question 3 above:

```
volatile unsigned long *cnt = 0x8;
volatile unsigned long *out = 0x200;
...
     unsigned long before, after;
     before = *cnt;
     .... do some work ...
     after = *cnt;
     *(out + 1) = (after - before);
```

The value at the special address 0x8 is the current simulation cycle; the output port at address 0x200 + 1 word prints an integer value to the simulator's output.

1. Add this mechanism to your program from question 3 to measure the time needed to draw the triangle.
2. Using this mechanism, measure the time needed to draw the triangle, using the default settings in `minisim.ini`. Then run again with `-o IOBUS0:Freq=1`. In both cases, express the throughput of your program in pixels by second. Explain your results.