Project 2 Report

Christina Pavlopoulou
cpavl001@ucr.edu

Andres Calderon
acald013@ucr.edu

May 23, 2016

1 DVFS controller implementation

In order to implement the requested controller we start capturing the user parameters for the number of cycles per interval (DVFSInterval) and the target power (DVFSTargetPower) in the file `sim-outorder.c`. We did that using the `opt_reg_*` functions already provided by the code (listing 1). With these values we implement the controller as it is shown in listing 3. Listing 2 shows the variable we used.

In addition, we did modifications on files `power.c` and `power.h` to adapt the formula for power factor according to the changes in VSF and FSF values (listing 4). Also, we capture the total power per cycle for each interval at the end of the `update_power_stats` function (listing 5).

```
/*******************************************/
/* CS 203A Catching VSF and FSF from user */
/*******************************************/
  opt_reg_int(odb, "-DVFSInterval", "Number of cycles for power monitoring interval",
         &DVFSInterval, /* default */100000,
         /* print */TRUE, /* format */NULL);

  opt_reg_float(odb, "-DVFSTargetPower", "Target power budget controlled at each interval",
         &DVFSTargetPower, /* default */6000000.00,
         /* print */TRUE, /* format */"%12.2f");

  opt_reg_float(odb, "-DVFSIncrement", "Increment for scaling of Voltage and Frequency",
         &DVFSIncrement, /* default */0.1,
         /* print */TRUE, /* format */"%12.1f");

  opt_reg_flag(odb, "-DVFSTurnOff", "Do not execute the DVFS controller",
         &DVFSTurnOff, /* default */FALSE,
         /* print */TRUE, /* format */NULL);

/*******************************************/
/*                                         */
/*******************************************/
```

Listing 1: Capturing user parameters.

```
/******************************************/
/* CS 203A Declaring VSF and FSF          */
/******************************************/
float VSF = 1.0;
float FSF = 1.0;

/******************************************/
/* CS 203A Declaring arguments            */
/******************************************/
int DVFSInterval;
float DVFSTargetPower;
float DVFSIncrement;
int DVFSTurnOff;

/******************************************/
/* CS 203A Declaring auxiliar varibles    */
/******************************************/
#define Mhz 600e6
float power_this_interval;
float total;
float previous_total = 0;
float avg_power;
float power_factor;
extern FILE * output;

/******************************************/
/*                                        */
/******************************************/
```

Listing 2: Declaration of variables.

```
/**********************************************************************************************/
/* CS 203A DVFS Controller                                                                    */
/**********************************************************************************************/

  if(sim_cycle % DVFSInterval == 0){
    power_this_interval = total - previous_total;
    avg_power = power_this_interval / DVFSInterval;

    if(DVFSTurnOff == FALSE){
      if(power_this_interval > DVFSTargetPower && VSF > 0.3){
        VSF -= DVFSIncrement;
        FSF -= DVFSIncrement;
      }
      if(power_this_interval < DVFSTargetPower && VSF < 10.0){
        VSF += DVFSIncrement;
        FSF += DVFSIncrement;
      }
    }

    fprintf(output,"%f:%f:%f:%f:%f\n", power_this_interval, avg_power, VSF, FSF, FSF*Mhz);
    previous_total += power_this_interval;
  }

/**********************************************************************************************/
/*                                                                                            */
/**********************************************************************************************/
```

Listing 3: Source code DVFS controller.

```c
/******************************************/
/* CS 203A Defining a new Power factor    */
/******************************************/

extern float VSF;
extern float FSF;
#define Powerfactor (FSF)*(Mhz)*(VSF*VSF)*Vdd*Vdd
extern float total;
extern float power_factor;

/******************************************/
/*                                        */
/******************************************/
```

Listing 4: Power factor modification.

```c
  /********************************************************************************/
  /* CS 203A Computing total power per cycle                                      */
  /********************************************************************************/
  total += total_cycle_power_cc1 + total_cycle_power_cc2 + total_cycle_power_cc3;

  /********************************************************************************/
  /*                                                                              */
  /********************************************************************************/
```

Listing 5: Computing total power per interval.

2 Experiments

To run the experiment we select target power at each case without having the controller working. For go, we found that the average total power consumption was around 5400000. In the case of Anagram the average power consumption was 7400000. We modified the upper bound of the scaling factor to 10 in order to get a more wide range of possible values.

2.1 Go

Given time restriction we limit the number of instruction for this experiment to 50 million using the `-max:inst` parameter. Table 1 summarizes the results of the experiment. When we use the controller, we can see that while the total and average power consumption is better, the execution time increases. Figures 1 and 2 show the change on the scaling factors and the corresponding total power consumption respectively at each interval. The red line on figure 2 marks the target power. We can see that the power remains around the target.

	DVFS Controller	Baseline
Total run-time	731 ns	610 ns
Total energy consumption	1998777579	2008539742
Average power consumption	53.75	54.02

Table 1: Results for the Go experiment

2.2 Anagram

We can see a similar behavior to the previous section. Table 2 and Figures 3 and 4 summarize the results for this experiment.

	DVFS Controller	Baseline
Total run-time	444 ns	418 ns
Total energy consumption	1880593561	1893120690
Average power consumption	74.33	74.82

Table 2: Results for the Anagram experiment

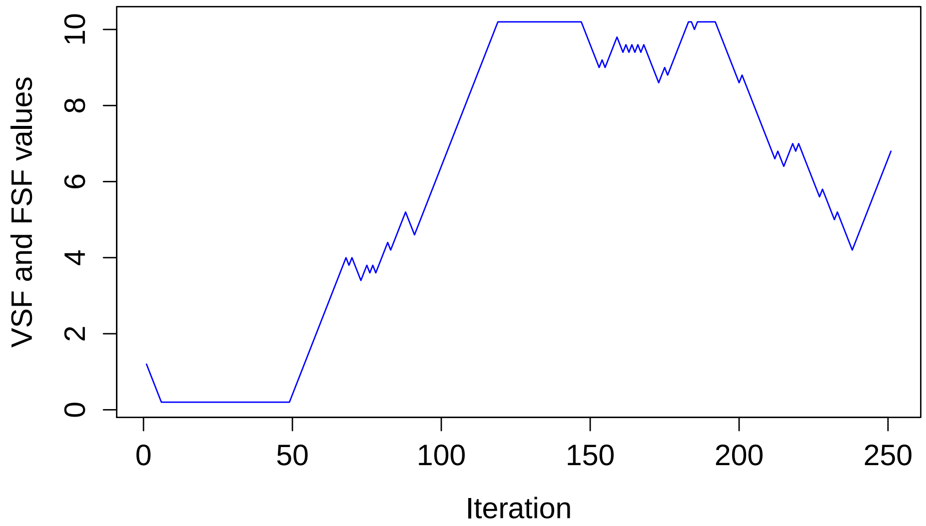

Figure 1: Scaling factors VSF and FSF for each interval for the Go experiment.

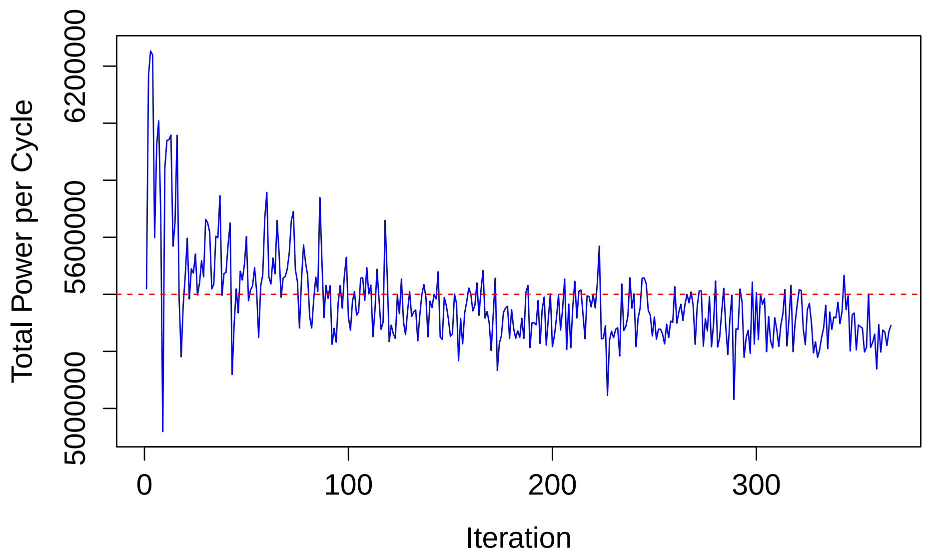

Figure 2: Total power per interval for the Go experiment. Red line is the target power.

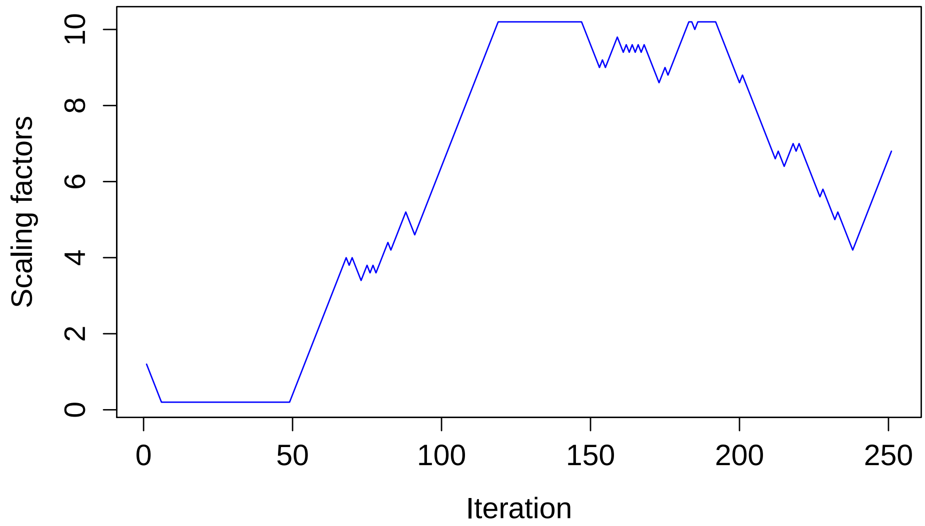

Figure 3: Scaling factors VSF and FSF for each interval for the Anagram experiment.

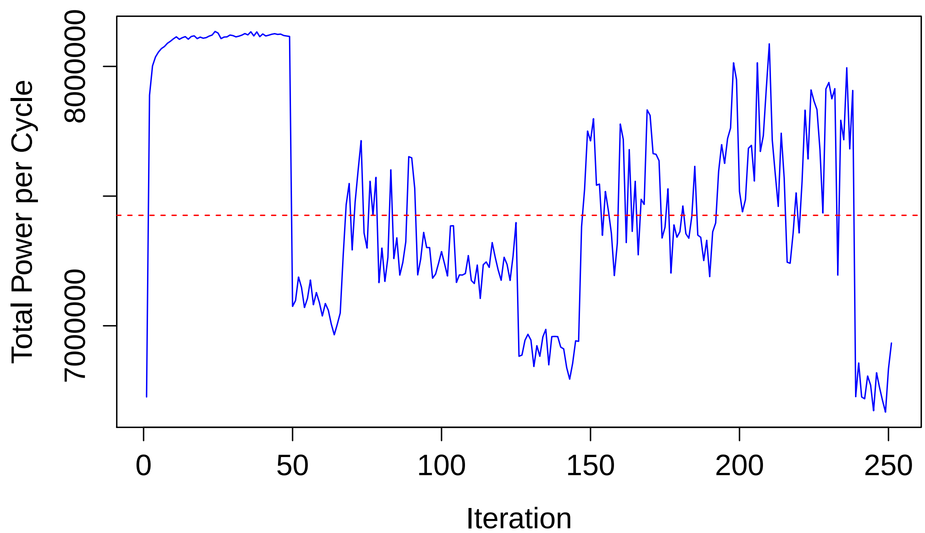

Figure 4: Total power per interval for the Anagram experiment. Red line is the target power.