OOP - Final Project

Hardware platform simulator

Prepared by

Cristiano MERIO | Robert NAKHLE

Supervised by

Mr. Andrea BATTISTELLA

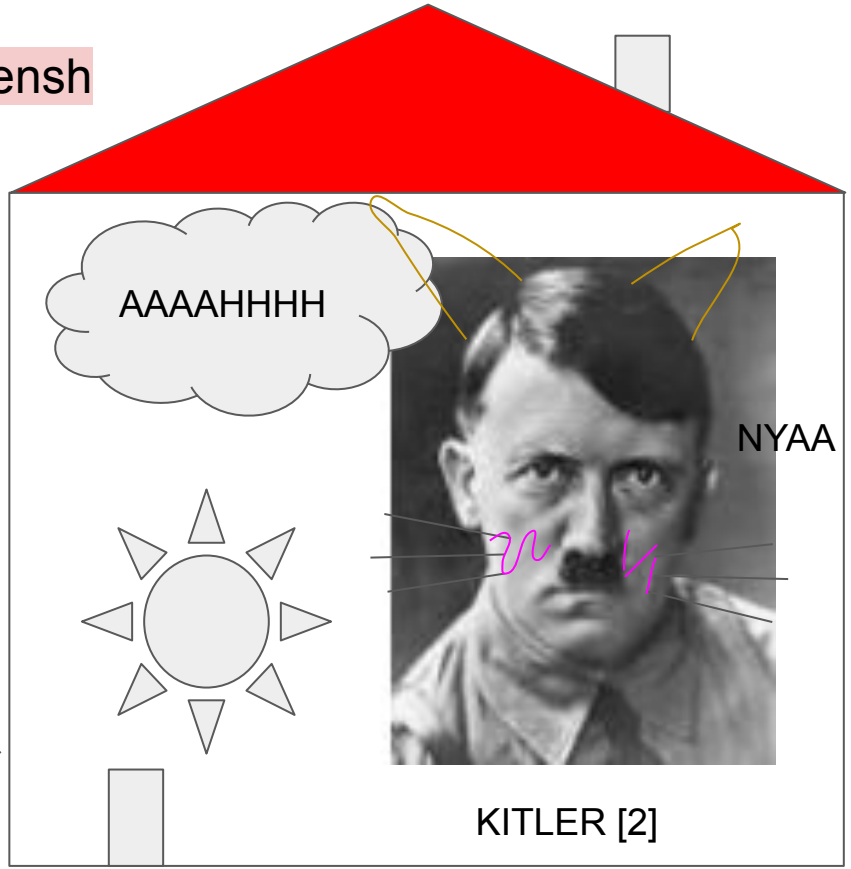
Plan

- Description des different block: FSM et testbensh
- Présentation du système et RTL
- Le main
- Les stimuli
- Remarques particulières

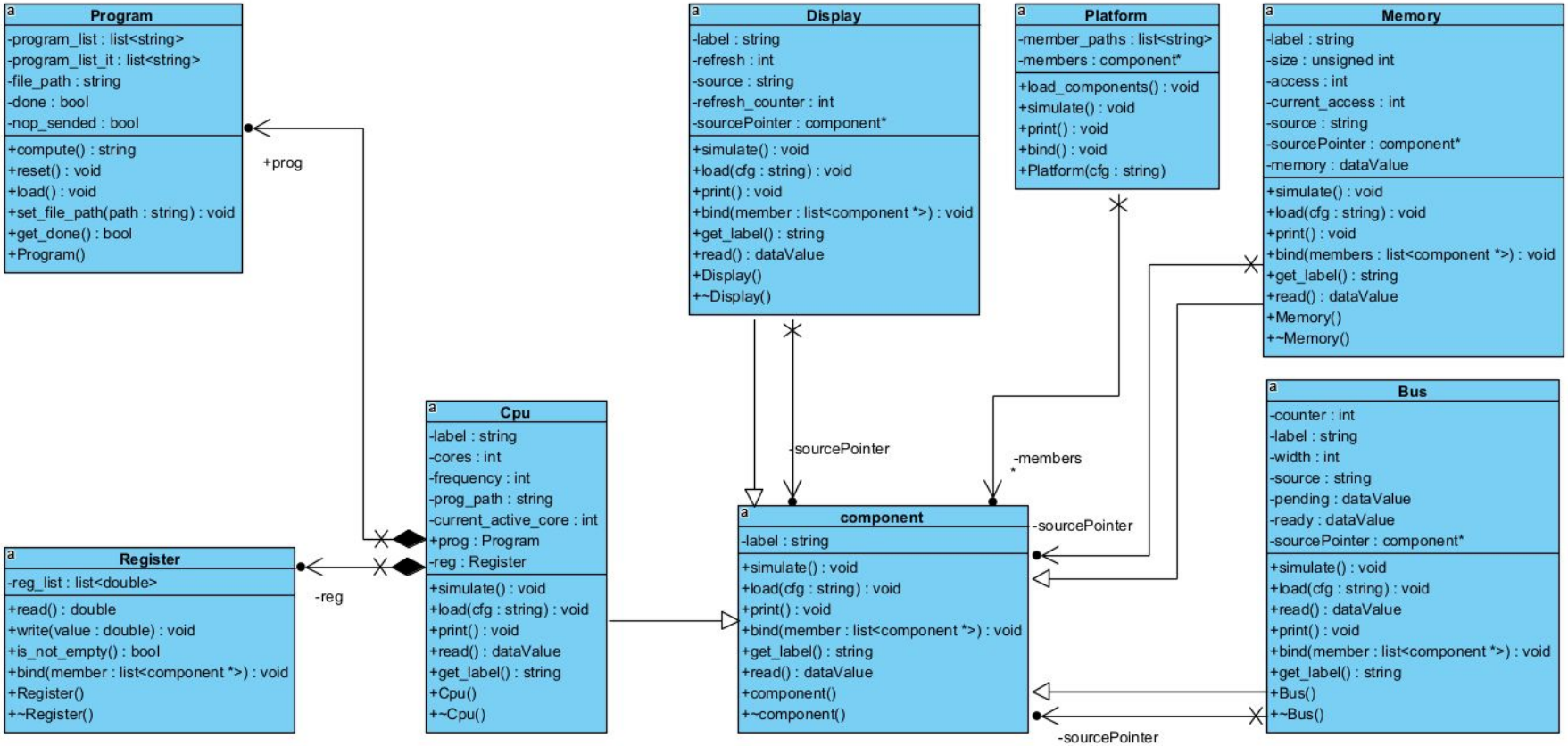
jhristiano merhhio

rouberths naglie

Hello



Code plan


Program
-program_list : list<string>
-program_list_it : list<string>
-file_path : string
-done : bool
-nop_sended : bool
+compute() : string
+reset() : void
+load() : void
+set_file_path(path : string) : void
+get_done() : bool
+Program()
+prog
Display
-label : string
-refresh : int
-source : string
-refresh_counter : int
-sourcePointer : component*
+simulate() : void
+load(cfg : string) : void
+print() : void
+bind(member : list<component *>) : void
+get_label() : string
+read() : dataValue
+Display()
+~Display()
Platform
-member_paths : list<string>
-members : component*
+load_components() : void
+simulate() : void
+print() : void
+bind() : void
+Platform(cfg : string)
Memory
-label : string
-size : unsigned int
-access : int
-current_access : int
-source : string
-sourcePointer : component*
-memory : dataValue
+simulate() : void
+load(cfg : string) : void
+print() : void
+bind(members : list<component *>) : void
+get_label() : string
+read() : dataValue
+Memory()
+~Memory()
Cpu
-label : string
-cores : int
-frequency : int
-prog_path : string
-current_active_core : int
+prog : Program
-reg : Register
+simulate() : void
+load(cfg : string) : void
+print() : void
+read() : dataValue
+get_label() : string
+Cpu()
+~Cpu()
-sourcePointer
-members
*
Bus
-counter : int
-label : string
-width : int
-source : string
-pending : dataValue
-ready : dataValue
-sourcePointer : component*
+simulate() : void
+load(cfg : string) : void
+read() : dataValue
+print() : void
+bind(member : list<component *>) : void
+get_label() : string
+Bus()
+~Bus()
Register
-reg_list : list<double>
+read() : double
+write(value : double) : void
+is_not_empty() : bool
+bind(member : list<component *>) : void
+Register()
+~Register()
-reg
component
-label : string
+simulate() : void
+load(cfg : string) : void
+print() : void
+bind(member : list<component *>) : void
+get_label() : string
+read() : dataValue
+component()
+~component()
-sourcePointer
-sourcePointer

component.hpp

```
class component {
public:
  virtual void simulate();
  virtual void load(string cfg);
  virtual void print();
  virtual void bind(list<component *> member);
  virtual string get_label();
  virtual dataValue read();

  // constructors
  component();
  virtual ~component();

private:
  string label;
};
```

platform.hpp

```
class Platform {
public:
  void load_components();
  void simulate();
  void print();
  void bind();

  // constructors
  Platform(string cfg);

private:
  list<component *> members;
  list<string> member_paths;
};
```

bus.hpp

```
class Bus : virtual public component {
public:
  void simulate();
  void load(string cfg);
  dataValue read();
  void print();
  void bind(list<component *> member);
  string get_label();

  // constructors
  Bus();
  ~Bus();

private:
  list<dataValue> pending;
  list<dataValue> ready;
  int counter;

  string label;
  component *sourcePointer;
  int width;
  string source;
};
```

Cpu.hpp

```
class Cpu : virtual public component {
public:
  void simulate();
  void load(string cfg);
  void print();
  // TODO
  dataValue read();
  string get_label();

  // constructors
  Cpu();
  ~Cpu();

  Program prog; // TODO return private
private:
  string label;
  int cores;
  int frequency;
  string prog_path;
  int current_active_core;

  Register reg;
};
```

Program.hpp - Register.hpp

```
class Program {
public:
  string compute();
  void reset();
  void load();
  void set_file_path(string path);
  bool get_done();
  // constructors
  Program();

private:
  list<string> program_list;
  list<string>::iterator program_list_it;

  string file_path;
  bool done;
  bool nop_sended;
};
```

```
class Register {
public:
  double read();
  void write(double value);
  bool is_not_empty();
  void bind(list<component *> member);

  // constructors
  Register();
  ~Register();

private:
  list<double> reg_list;
};
```

Display.hpp

```
class Display : virtual public component {
public:
  void simulate();
  void load(string cfg);
  void print();
  void bind(list<component *> member);
  string get_label();
  dataValue read();

  // constructors
  Display();
  ~Display();

private:
  string label;
  int refresh;
  string source;

  component *sourcePointer;
  int refresh_counter;
};
```

Memory.hpp

```
class Memory : virtual public component {
public:
  void simulate();
  void load(string cfg);
  void print();
  void bind(list<component *> members);
  string get_label();
  dataValue read();

  // constructors
  Memory();
  ~Memory();

private:
  component * sourcePointer;
  string label;
  unsigned int size;
  int access;
  int current_access;
  string source;
  list<dataValue> memory;
};
```

dataValue - option_t

```
typedef struct {
  double value;
  bool flag;
} dataValue;
```

```
typedef struct {
  string field;
  string data;
} option_t;
```

file.hpp - debug.h

```
list<string> file_to_list(string);
list<string> string_to_words(string);
list<option_t> parse_cfg(string);
string find_in_cfg(string, list<option_t>);

bool string_contains(string, string);

string rtrim(string);
string ltrim(string);
string trim(string);
```

```
#ifndef DEBUG_H
#define DEBUG_H
#define DEBUG(x) do { \
  if (IS_DEBUG_ON) { std::cerr << x << std::endl; } \
} while (0)
#endif
```

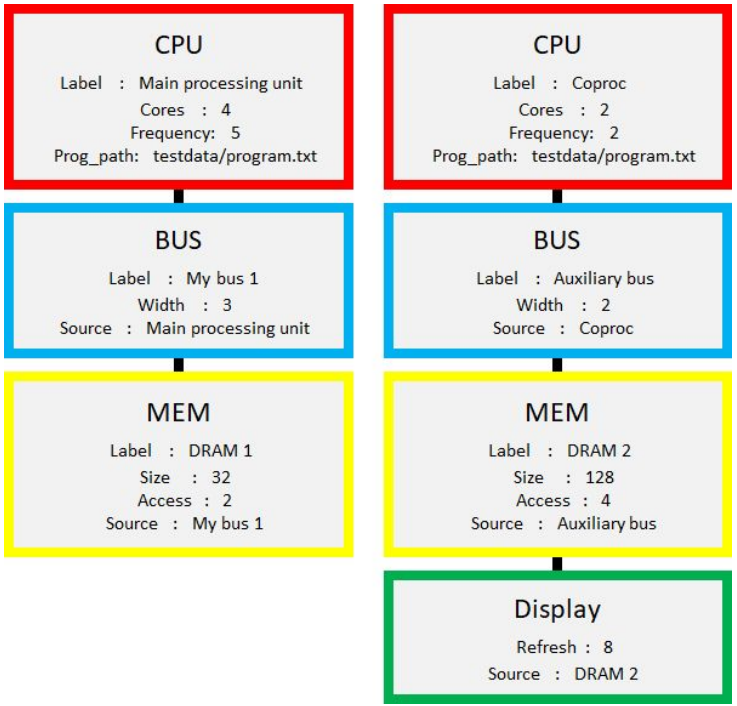
Le main

- Création d'une nouvelle plateforme
- Load + Bind
- Print des membres de la simulation
- Choix du fichier plateforme utilisé
- Simulation selon le nombre de cycles entré par l'utilisateur

Le main

```cpp
#include "platform.hpp"

const string directory = "./";

using namespace std;

int main(int argc, char *argv[]) {

  if(argc != 2){
      cerr << "Invalid nb of arguments" << endl;
      exit(1);
  }

  Platform myPlatform("testdata/platform.txt");
  myPlatform.load_components();
  myPlatform.print();
  myPlatform.bind();

  int sim = stoi(argv[1]);

  for (int kitler = 1; kitler < sim; kitler++) {
    myPlatform.simulate();
  }

  exit(0);

  return 0;
}
```

TestData


CPU
Label : Main processing unit
Cores : 4
Frequency: 5
Prog_path: testdata/program.txt
BUS
Label : My bus 1
Width : 3
Source : Main processing unit
MEM
Label : DRAM 1
Size : 32
Access : 2
Source : My bus 1
CPU
Label : Coproc
Cores : 2
Frequency: 2
Prog_path: testdata/program.txt
BUS
Label : Auxiliary bus
Width : 2
Source : Coproc
MEM
Label : DRAM 2
Size : 128
Access : 4
Source : Auxiliary bus
Display
Refresh : 8
Source : DRAM 2

TestData

```
Initialising platform..
Loading components...
Printing the platform's content!
[Memory Print]
   Label     :   DRAM 1
   Size      :   32
   Access    :   2
   Source    :   My bus 1

[Memory Print]
   Label     :   DRAM 2
   Size      :   128
   Access    :   4
   Source    :   Auxiliary bus

[Bus Print]
   Label     :   My bus 1
   Width     :   3
   Source    :   Main processing unit

[Bus Print]
   Label     :   Auxiliary bus
   Width     :   2
   Source    :   Coproc
```

```
[CPU Print]
   Label     :   Main processing unit
   Cores     :   4
   Frequency:    5
   Prog_path:    testdata/program.txt

[CPU Print]
   Label     :   Coproc
   Cores     :   2
   Frequency:    2
   Prog_path:    testdata/program.txt

[Display Print]
   Refresh   :   8
   Source    :   DRAM 2

Done printing content!
Starting platform bind!
Platform bind complete.
```

TestData

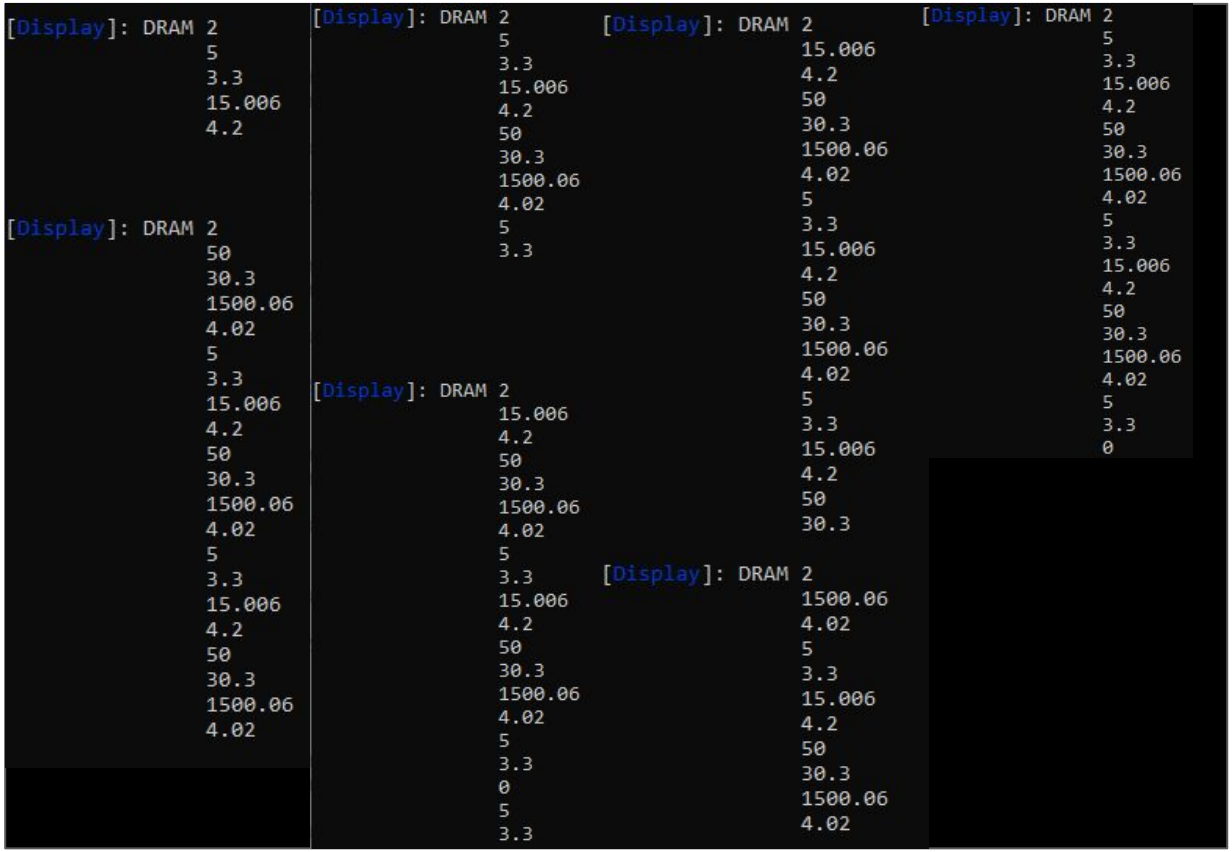
```
[Display]: DRAM 2
5
3.3
15.006
4.2

[Display]: DRAM 2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
```

```
[Display]: DRAM 2
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3

[Display]: DRAM 2
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
0
5
3.3
```

```
[Display]: DRAM 2
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3

[Display]: DRAM 2
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
```

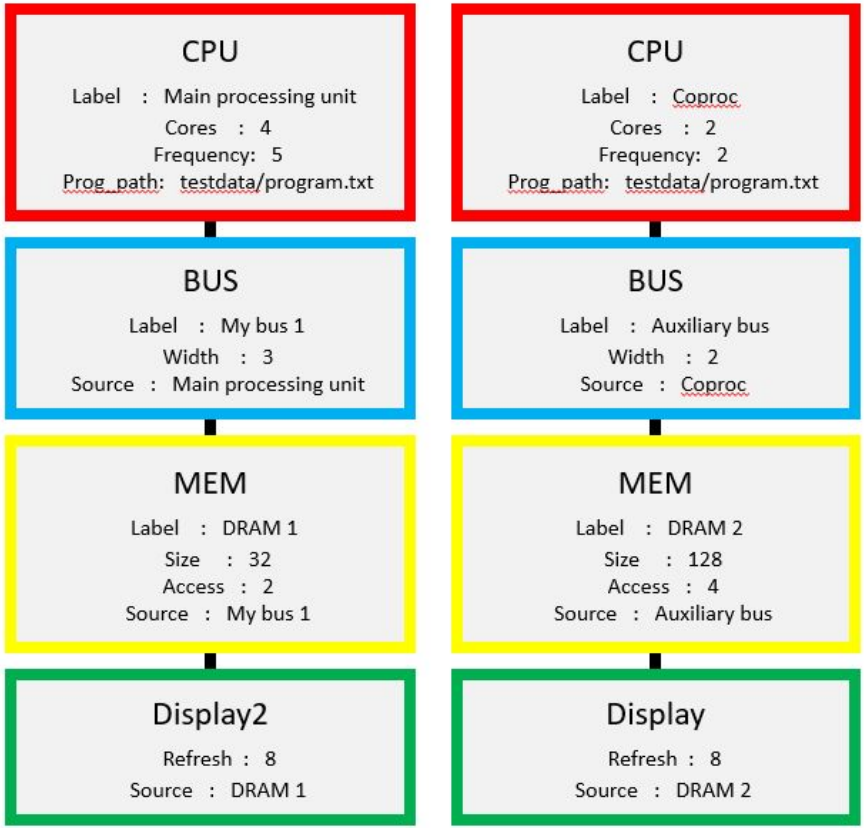
```
[Display]: DRAM 2
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
0
```

TestData




```
Initialising platform..
Loading components...
Printing the platform's content!
[Memory Print]
   Label    :   DRAM 1
   Size     :   32
   Access   :   2
   Source   :   My bus 1

[Memory Print]
   Label    :   DRAM 2
   Size     :   128
   Access   :   4
   Source   :   Auxiliary bus

[Bus Print]
   Label    :   My bus 1
   Width    :   3
   Source   :   Main processing unit

[Bus Print]
   Label    :   Auxiliary bus
   Width    :   2
   Source   :   Coproc

[CPU Print]
   Label    :   Main processing unit
   Cores    :   4
   Frequency:   5
   Prog_path:   testdata/program.txt

[CPU Print]
   Label    :   Coproc
   Cores    :   2
   Frequency:   2
   Prog_path:   testdata/program.txt

[Display Print]
   Refresh  :   8
   Source   :   DRAM 2

Done printing content!
Starting platform bind!
Platform bind complete.
```

TestData 2


CPU
Label : Main processing unit
Cores : 4
Frequency: 5
Prog_path: testdata/program.txt
BUS
Label : My bus 1
Width : 3
Source : Main processing unit
MEM
Label : DRAM 1
Size : 32
Access : 2
Source : My bus 1
Display2
Refresh : 8
Source : DRAM 1
CPU
Label : Coproc
Cores : 2
Frequency: 2
Prog_path: testdata/program.txt
BUS
Label : Auxiliary bus
Width : 2
Source : Coproc
MEM
Label : DRAM 2
Size : 128
Access : 4
Source : Auxiliary bus
Display
Refresh : 8
Source : DRAM 2

TestData 2

```
Initialising platform..
Loading components...
Printing the platform's content!
[Memory Print]
   Label     :   DRAM 1
   Size      :   32
   Access    :   2
   Source    :   My bus 1

[Memory Print]
   Label     :   DRAM 2
   Size      :   128
   Access    :   4
   Source    :   Auxiliary bus

[Bus Print]
   Label     :   My bus 1
   Width     :   3
   Source    :   Main processing unit

[Bus Print]
   Label     :   Auxiliary bus
   Width     :   2
   Source    :   Coproc
```

```
[CPU Print]
   Label     :   Main processing unit
   Cores     :   4
   Frequency:    5
   Prog_path:    testdata/program.txt

[CPU Print]
   Label     :   Coproc
   Cores     :   2
   Frequency:    2
   Prog_path:    testdata/program.txt

[Display Print]
   Refresh   :   8
   Source    :   DRAM 2

[Display Print]
   Refresh   :   8
   Source    :   DRAM 1

Done printing content!
Starting platform bind!
Platform bind complete.
```

TestData 2

```
[Display]: DRAM 2
5
3.3
15.006
4.2

[Display]: DRAM 1
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5


[Display]: DRAM 2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
```

```
[Display]: DRAM 1
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2


[Display]: DRAM 2
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
```

```
[Display]: DRAM 1
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
0
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2


[Display]: DRAM 2
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
0
5
3.3
```

```
[Display]: DRAM 1
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3


[Display]: DRAM 2
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
```

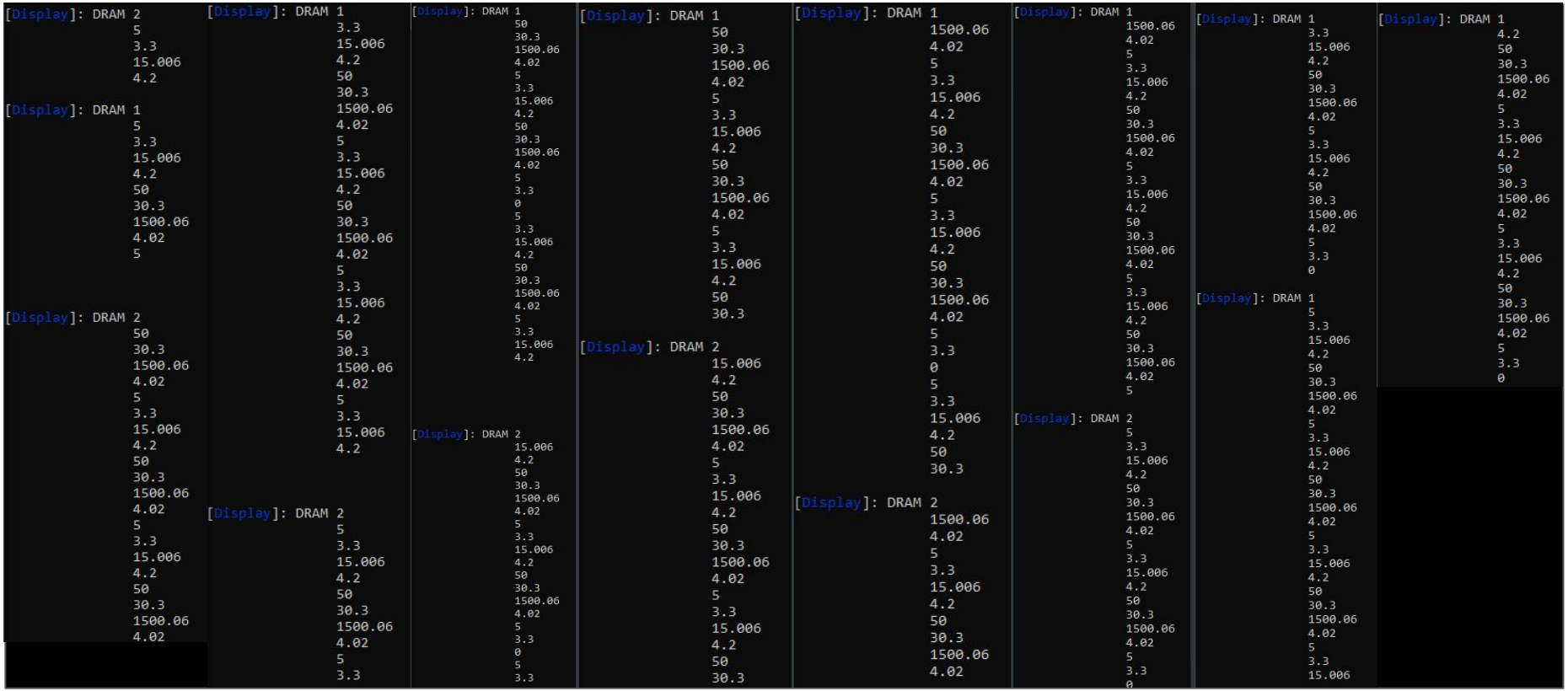
```
[Display]: DRAM 1
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
0
5
3.3
15.006
4.2
50
30.3


[Display]: DRAM 2
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
```

```
[Display]: DRAM 1
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5


[Display]: DRAM 2
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
0
```

```
[Display]: DRAM 1
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
0


[Display]: DRAM 1
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
```

```
[Display]: DRAM 1
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
0
```

TestData 2


[Display]: DRAM 2
5
3.3
15.006
4.2
[Display]: DRAM 1
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
[Display]: DRAM 2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
[Display]: DRAM 1
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
[Display]: DRAM 2
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
[Display]: DRAM 1
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
0
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
[Display]: DRAM 2
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
[Display]: DRAM 1
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
[Display]: DRAM 2
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
0
5
3.3
15.006
4.2
50
30.3
[Display]: DRAM 1
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
0
5
3.3
15.006
4.2
50
30.3
[Display]: DRAM 2
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
[Display]: DRAM 1
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
0
[Display]: DRAM 2
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
[Display]: DRAM 1
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
0
[Display]: DRAM 1
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
[Display]: DRAM 1
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
15.006
4.2
50
30.3
1500.06
4.02
5
3.3
0

OOP - Final Project

Hardware platform simulator

Prepared by

Cristiano MERIO | Robert NAKHLE

Supervised by

Mr. Andrea BATTISTELLA