Selected MaxCompiler Examples

Sasa Stojanovic
stojsasa@etf.rs

Veljko Milutinovic
vm@etf.rs

How-to? What-to?

- One has to know
  how to program DataFlow machines,
  in order to get
  the best possible speedup out of them!

- For some applications (G),
  there is a large difference between
  what an experienced programmer achieves,
  and
  what an un-experienced one can achieve!

- For some other applications (B),
  no matter how experienced the programmer is,
  the speedup will not be revolutionary
  (may be even <1).

Lemas

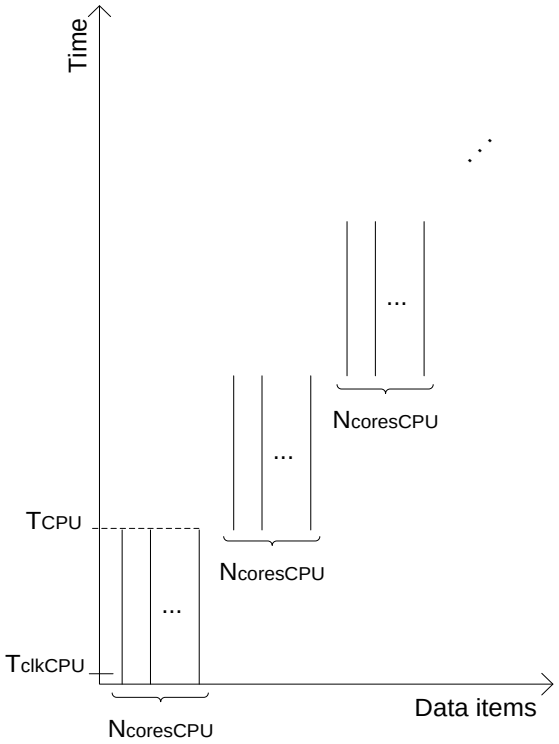
- Lemas:
  - 1. The how-to and how-not-to is important to know!
  - 2. The what-to and what-not-to is important to know!

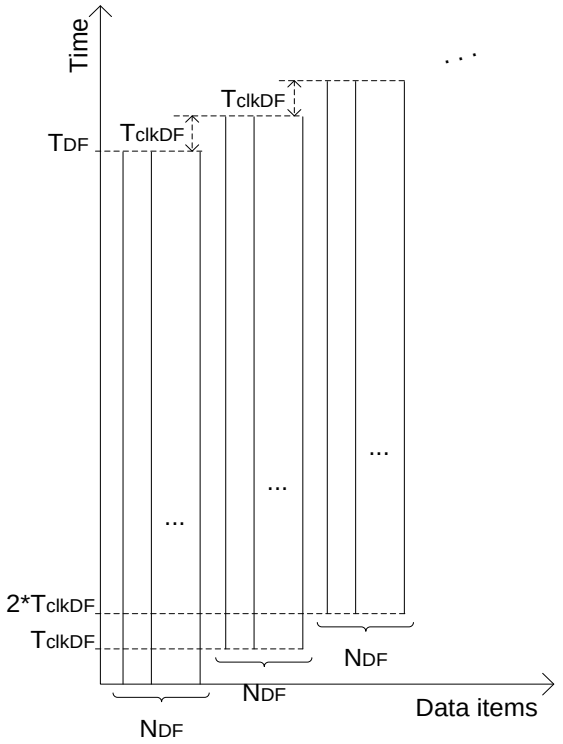
- N.B.
  - The how-to is taught through most of the examples to follow (all except the introductory ones).
  - The what-to/what-not-to is taught using a figure.

The Essential Figure:

$t_{CPU} = N * N_{OPS} * C_{CPU} * T_{clkCPU} / N_{coresCPU}$

$t_{DF} = N_{OPS} * C_{DF} * T_{clkDF} + (N - 1) * T_{clkDF} / N_{DF}$





Assumptions:

1. Software includes enough parallelism to keep all cores busy
2. The only limiting factor is the number of cores.

Bottomline:

- When is DataFlow better?
  - If
    the number of operations in a single loop iteration
    is above some critical value
    ADDITIVE SPEEDUP ENABLER
  - Then
    More data items means more advantage for DataFlow.
    ADDITIVE SPEEDUP MAKER
- In other words:
  - More data does not mean better performance
    if the #operations/iteration is below a critical value.
- Conclusion:
  - If we see an application with a small #operations/iteration,
    it is possibly (not always) a “what-not-to” application,
    and we better execute it on the host;
    otherwise, we will (or may) have a slowdown.

To have it more concrete:

- DataFlow: One new result in each cycle
  e.g. Clock = 100MHz
  Period = 10ns
  One result every 10ns
  [No matter how many operations in each loop iteration]

  Consequently: More operations does not mean proportionally more time; however, more operations means higher latency till the first result.

- MultiCore: One new result after each iteration
  e.g. Clock=4GHz
  Period = 250ps
  One result every 250ps times #ops
  [If #ops > 40 => DataFlow is better, although it uses a slower clock]

- Also: The MltiCore example will feature an additional slowdown, due to memory hierarchy access and pipeline related hazards
  =>
  critical #ops (bringing the same performance) is significantly below 40!!!

Don't missunderstand!

- DataFlow has no cache,
  but does have a memory hierarchy.

- However,
  memory hierarchy access with DataFlow
  is carefully planed
  by the programmer
  at the program write time

- As opposed to
  memory hierarchy access
  with a MultiCore
  which calculates the access address
  at the program run time.

N.B.

- Java to configure Maxeler DataFlow!
  C to program the host!

- One or more kernels!
  Only one manager!

- In theory,
  Simulator builder not needed
  if a card is used.
  In practice,
  you need it until the testing is over,
  since the compilation process is slow, for hardware,
  and fast, for software (simulator).

Example No.1: Hello World!

- Write a program that sends the “Hello World!” string to the MAX2 card, for the MAX2 card kernel to return it back to the host.

- To be learned through this example:
  - How to make the configuration of the accelerator (MAX2 card) using Java:
    - How to make a simple kernel (ops description) using Java (the only language),
    - How to write the standard manager (config description based on kernel(s)) using Java,
  - How to test the kernel using a test (code+data) written in Java,
  - How to compile the Java code for MAX2,
  - How to write a simple C code that runs on the host and triggers the kernel,
    - How to write the C code that streams data to the kernel,
    - How to write the C code that accepts data from the kernel,
  - How to simulate and execute an application program in C that runs on the host and periodically calls the accelerator.

Standard Files in a MAX Project

- One or more kernel files, to define operations of the application:
  - <app_name>Kernel[<additional_name>].java
- One (or more) Java file, for simulation of the kernel(s):
  - <app_name>SimRunner.java
- One manager file for transforming the kernel(s) into the configuration of the MAX card (instantiation and connection of kernels):
  - <app_name>Manager.java
- Simulator builder:
  - <app_name>HostSimBuilder.java
- Hardware builder:
  - <app_name>HWBuilder.java
- Application code that uses the MAX card accelerator:
  - <app_name>HostCode.c
- Makefile
  - A script file that defines the compilation related commands

example1Kernel.java

```
package ind.z1;

import com.maxeler.maxcompiler.v1.kernelcompiler.Kernel;
import com.maxeler.maxcompiler.v1.kernelcompiler.KernelParameters;
import com.maxeler.maxcompiler.v1.kernelcompiler.types.base.HWVar;

public class helloKernel extends Kernel {
    public helloKernel(KernelParameters parameters) {
        super(parameters);
// Input:
        HWVar x = io.input("x", hwInt(8));
        HWVar result = x;
// Output:
        io.output("z", result, hwInt(8));
    }
}
```

It is possible to substitute the last three lines with:
io.output("z", x, hwInt(8));

example1SimRunner.java

```
package ind.z1;

import com.maxeler.maxcompiler.v1.managers.standard.SimulationManager;

public class helloSimRunner {
    public static void main(String[] args) {
            SimulationManager m = new SimulationManager(“helloSim");
            helloKernel k = new helloKernel( m.makeKernelParameters() );
            m.setKernel(k);
            m.setInputData("x", 1, 2, 3, 4, 5, 6, 7, 8);
            m.setKernelCycles(8);
            m.runTest();
            m.dumpOutput();
            double expectedOutput[] = { 1, 2, 3, 4, 5, 6, 7, 8 };
            m.checkOutputData("z", expectedOutput);
            m.logMsg("Test passed OK!");
    }
}
```

example1HostSimBuilder.java

```
package ind.z1;

import static config.BoardModel.BOARDMODEL;
import com.maxeler.maxcompiler.v1.kernelcompiler.Kernel;
import com.maxeler.maxcompiler.v1.managers.standard.Manager;
import com.maxeler.maxcompiler.v1.managers.standard.Manager.IOType;

public class helloHostSimBuilder {
    public static void main(String[] args) {
        Manager m = new Manager(true,”helloHostSim", BOARDMODEL);
        Kernel k = new
                helloKernel(m.makeKernelParameters(“helloKernel"));
        m.setKernel(k);
        m.setIO(IOType.ALL_PCIE);
        m.build();
    }
}
```

example1HwBuilder.java

```
package ind.z1;

import static config.BoardModel.BOARDMODEL;
import com.maxeler.maxcompiler.v1.kernelcompiler.Kernel;
import com.maxeler.maxcompiler.v1.managers.standard.Manager;
import com.maxeler.maxcompiler.v1.managers.standard.Manager.IOType;

public class helloHWBuilder {
    public static void main(String[] args) {
        Manager m = new Manager("hello", BOARDMODEL);
        Kernel k = new helloKernel( m.makeKernelParameters() );
        m.setKernel(k);
        m.setIO(IOType.ALL_PCIE);
        m.build();
    }
}
```

example1HostCode.c 1/2

```
#include <stdio.h>
#include <MaxCompilerRT.h>

int main(int argc, char* argv[])
{
    char *device_name = (argc==2 ? argv[1] : "/dev/maxeler0");
    max_maxfile_t* maxfile;
    max_device_handle_t* device;
    char data_in1[16] = "Hello world!";
    char data_out[16];

    printf("Opening and configuring FPGA.\n");

    maxfile = max_maxfile_init_hello();
    device = max_open_device(maxfile, device_name);
    max_set_terminate_on_error(device);
```

example1HostCode.c 2/2

```
    printf("Streaming data to/from FPGA...\n");

    max_run(device,
                max_input("x", data_in1, 16 * sizeof(char)),
                max_output("z", data_out, 16 * sizeof(char)),
                max_runfor("helloKernel", 16),
                max_end());

    printf("Checking data read from FPGA.\n");

    max_close_device(device);
    max_destroy(maxfile);

    return 0;
}
```

Makefile

```
# Root of the project directory tree
BASEDIR=../../..
# Java package name
PACKAGE=ind/z1
# Application name
APP=example1
# Names of your maxfiles
HWMAXFILE=$(APP).max
HOSTSIMMAXFILE=$(APP)HostSim.max
# Java application builders
HWBUILDER=$(APP)HWBuilder.java
HOSTSIMBUILDER=$(APP)HostSimBuilder.java
SIMRUNNER=$(APP)SimRunner.java
# C host code
HOSTCODE=$(APP)HostCode.c
# Target board
BOARD_MODEL=23312
# Include the master makefile.include
nullstring :=
space := $(nullstring) # comment
MAXCOMPILERDIR_QUOTE:=$(subst $(space),\ ,$(MAXCOMPILERDIR))
include $(MAXCOMPILERDIR_QUOTE)/examples/common/Makefile.include
```

BoardModel.java

```
package config;

import
   com.maxeler.maxcompiler.v1.managers.MAX2BoardModel;

public class BoardModel {
   public static final MAX2BoardModel BOARDMODEL =
         MAX2BoardModel.MAX2336B;
}
```

Example No. 2: Vector Addition

- Write a program
  that adds two arrays
  of floating point numbers.
- Program reads the size of arrays,
  makes two arrays
  with an arbitrary content (test inputs),
  and adds them using a MAX card.

example2Kernel.Java

```
package ind.z2;

import com.maxeler.maxcompiler.v1.kernelcompiler.Kernel;
import com.maxeler.maxcompiler.v1.kernelcompiler.KernelParameters;
import com.maxeler.maxcompiler.v1.kernelcompiler.types.base.HWVar;

public class example2Kernel extends Kernel {

    public example2Kernel(KernelParameters parameters) {
        super(parameters);

        // Input
        HWVar x = io.input("x", hwFloat(8,24));
        HWVar y = io.input("y", hwFloat(8,24));

        HWVar result = x + y;

        // Output
        io.output("z", result, hwFloat(8,24));

    }
}
```

example2SimRunner.java

```
package ind.z2;
import com.maxeler.maxcompiler.v1.managers.standard.SimulationManager;

public class example2SimRunner {

    public static void main(String[] args) {
            SimulationManager m = new SimulationManager("example2Sim");
            example2Kernel k = new example2Kernel( m.makeKernelParameters() );
            m.setKernel(k);

            m.setInputData("x", 1, 2, 3, 4, 5, 6, 7, 8);
            m.setInputData("y", 2, 3, 4, 5, 6, 7, 8, 9);
            m.setKernelCycles(8);

            m.runTest();

            m.dumpOutput();
            double expectedOutput[] = { 3, 5, 7, 9, 11, 13, 15, 17 };

            m.checkOutputData("z", expectedOutput);
            m.logMsg("Test passed OK!");
    }
}
```

example2HostSimBuilder.java

```
package ind.z2;

import static config.BoardModel.BOARDMODEL;

import com.maxeler.maxcompiler.v1.kernelcompiler.Kernel;
import com.maxeler.maxcompiler.v1.managers.standard.Manager;
import com.maxeler.maxcompiler.v1.managers.standard.Manager.IOType;

public class example2HostSimBuilder {

    public static void main(String[] args) {
            Manager m = new Manager(true,"example2HostSim", BOARDMODEL);
            Kernel k = new example2Kernel( m.makeKernelParameters("example2Kernel") );

            m.setKernel(k);

            m.setIO(IOType.ALL_PCIE);

            m.build();
    }
}
```

example2HWBuilder.java

```
package ind.z2;

import static config.BoardModel.BOARDMODEL;

import com.maxeler.maxcompiler.v1.kernelcompiler.Kernel;
import com.maxeler.maxcompiler.v1.managers.standard.Manager;
import com.maxeler.maxcompiler.v1.managers.standard.Manager.IOType;

public class example2HWBuilder {

    public static void main(String[] args) {
            Manager m = new Manager("example2", BOARDMODEL);
            Kernel k = new example2Kernel( m.makeKernelParameters() );

            m.setKernel(k);

            m.setIO(IOType.ALL_PCIE);

            m.build();
    }
}
```

example2HostCode.c 1/2

```
#include <stdio.h>
#include <stdlib.h>

#include <MaxCompilerRT.h>

int main(int argc, char* argv[])
{
	char *device_name = (argc==2 ? argv[1] : "/dev/maxeler0");
	max_maxfile_t* maxfile;
	max_device_handle_t* device;
	float *data_in1, *data_in2, *data_out;
	unsigned long N, i;

	printf("Enter size of array: ");  scanf("%lu",&N);
	data_in1 = malloc(N * sizeof(float));
	data_in2 = malloc(N * sizeof(float));
	data_out = malloc(N * sizeof(float));

	for(i = 0; i < N; i++){
		data_in1[i] = i%10;
		data_in2[i] = i%3;
	}

	printf("Opening and configuring FPGA.\n");
```

example2HostCode.c 2/2

```
    maxfile = max_maxfile_init_example2();
    device = max_open_device(maxfile, device_name);
    max_set_terminate_on_error(device);

    printf("Streaming data to/from FPGA...\n");
    max_run(device,
                        max_input("x", data_in1, N * sizeof(float)),
                        max_input("y", data_in2, N * sizeof(float)),
                        max_output("z", data_out, N * sizeof(float)),
                        max_runfor("example2Kernel", N),
                        max_end());

    printf("Checking data read from FPGA.\n");

    for(i = 0; i < N; i++)
        if (data_out[i] != i%10 + i%3){
                        printf("Error on element %d. Expected %f, but found %f.", i, (float)(i%10+i%3), data_out[i]);
                        break;
        }

    max_close_device(device);
    max_destroy(maxfile);
    return 0;
}
```

Example No. 3: Optimized Array Summation

- Write an optimized program that calculates the sum of numbers in an input array
- First, calculate several parallel/partial sums; then, add them at the end

example3Kernel1 .java

```
package ind.z3;

import com.maxeler.maxcompiler.v1.kernelcompiler.Kernel;
import com.maxeler.maxcompiler.v1.kernelcompiler.KernelParameters;
import com.maxeler.maxcompiler.v1.kernelcompiler.types.base.HWVar;
import com.maxeler.maxcompiler.v1.kernelcompiler.types.base.HWType;

public class example30Kernel1 extends Kernel {
    public example3Kernel1(KernelParameters parameters) {
            super(parameters);
            final HWType scalarType = hwFloat(8,24);
            HWVar cnt = control.count.simpleCounter(64);

            // Input
            HWVar N = io.scalarInput("N", hwUInt(64));
            HWVar x = io.input("x", hwFloat(8,24) );
            HWVar sum = scalarType.newInstance(this);

            HWVar result = x + (cnt>0?sum:0.0);
            sum <== stream.offset(result, -13);

            // Output
            io.output("z", result, hwFloat(8,24), cnt > N-14);
    }
}
```

example3Kernel2.java

```
package ind.z3;
import com.maxeler.maxcompiler.v1.kernelcompiler.Kernel;
import com.maxeler.maxcompiler.v1.kernelcompiler.KernelParameters;
import com.maxeler.maxcompiler.v1.kernelcompiler.types.base.HWVar;
import com.maxeler.maxcompiler.v1.kernelcompiler.types.base.HWType;
import com.maxeler.maxcompiler.v1.kernelcompiler.stdlib.core.CounterChain;

public class example3Kernel2 extends Kernel {
    public example3Kernel2(KernelParameters parameters) {
            super(parameters);
            final HWType scalarType = hwFloat(8,24);
            CounterChain cc = control.count.makeCounterChain();
            HWVar cnt = cc.addCounter(14,1);
            HWVar depth = cc.addCounter(13,1);

            // Input
            HWVar x = io.input("x", hwFloat(8,24), depth.eq(0) );
            HWVar sum = scalarType.newInstance(this);

            HWVar result = x + (cnt>0?sum:0.0);
            sum <== stream.offset(result, -13);

            // Output
            io.output("z", result, hwFloat(8,24), cnt.eq(12));
    }
}
```

example3SimRunner.java

```
package ind.z3;

import com.maxeler.maxcompiler.v1.managers.standard.SimulationManager;

public class example3SimRunner {

    public static void main(String[] args) {
        SimulationManager m = new SimulationManager("example3Sim");
        example3Kernel1 k = new example3Kernel1( m.makeKernelParameters() );

        m.setKernel(k);
        m.setInputData("x", 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26);
        m.setKernelCycles(26);

        m.runTest();

        m.dumpOutput();
        double exOutput[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39  };

        m.checkOutputData("z", exOutput);
        m.logMsg("Test passed OK!");
    }
}
```

example3Manager.java

```
package ind.z3;

import com.maxeler.maxcompiler.v1.managers.custom.blocks.KernelBlock;
import com.maxeler.maxcompiler.v1.managers.custom.CustomManager;
import com.maxeler.maxcompiler.v1.managers.MAXBoardModel;

class example3Manager extends CustomManager {
    public example3Manager(boolean is_simulation, String name, MAXBoardModel board_model ){
            super(is_simulation, board_model, name);
            KernelBlock kb1 =
                addKernel(new example10Kernel1(makeKernelParameters("example10Kernel1")));
            KernelBlock kb2 =
                addKernel(new example10Kernel2(makeKernelParameters("example10Kernel2")));

            kb1.getInput("x") <== addStreamFromHost("x");
            kb2.getInput("x") <== kb1.getOutput("z");
            addStreamToHost("z") <== kb2.getOutput("z");
    }
}
```

example3HostSimBuilder.java

```
package ind.z3;

import static config.BoardModel.BOARDMODEL;

import com.maxeler.maxcompiler.v1.managers.BuildConfig;
import com.maxeler.maxcompiler.v1.managers.BuildConfig.Level;

public class example3HostSimBuilder {

    public static void main(String[] args) {
        example3Manager m =
                new example3Manager(true,"example3HostSim", BOARDMODEL);

        m.setBuildConfig(new BuildConfig(Level.FULL_BUILD));

        m.build();
    }
}
```

example3HWBuilder.java

```
package ind.z3;

import static config.BoardModel.BOARDMODEL;

import com.maxeler.maxcompiler.v1.kernelcompiler.Kernel;
import com.maxeler.maxcompiler.v1.managers.standard.Manager;
import com.maxeler.maxcompiler.v1.managers.standard.Manager.IOType;

public class example3HWBuilder {

    public static void main(String[] args) {
        example3Manager m =
            new example3Manager(false,"example10HostSim", BOARDMODEL);

        m.setBuildConfig(new BuildConfig(Level.FULL_BUILD));

        m.build();
    }
}
```

example3HostCode.c 1/2

```
#include <stdio.h>
#include <stdlib.h>

#include <MaxCompilerRT.h>


int main(int argc, char* argv[])
{
    char *device_name = (argc==2 ? argv[1] : "/dev/maxeler0");
    max_maxfile_t* maxfile;
    max_device_handle_t* device;
    float *data_in1, *data_out, expected = 0;
    unsigned long N, i;

    printf("Enter size of array (it will be truncated to the firs lower number dividable with 13): ");
    scanf("%lu",&N);
    N /= 13;
    N *= 13;
    data_in1 = malloc(N * sizeof(float));
    data_out = malloc(1 * sizeof(float));

    for(i = 0; i < N; i++){
            data_in1[i] = i%10;
            expected += data_in1[i];
    }
```

example3HostCode.c 2/2

```
    printf("Opening and configuring FPGA.\n");

    maxfile = max_maxfile_init_example3();
    device = max_open_device(maxfile, device_name);
    max_set_terminate_on_error(device);

    max_set_scalar_input_f(device, "example10Kernel1.N", N, FPGA_A);
    max_upload_runtime_params(device, FPGA_A);

    printf("Streaming data to/from FPGA...\n");
    max_run(device,
                        max_input("x", data_in1, N * sizeof(float)),
                        max_output("z", data_out, 2 * sizeof(float)),
                        max_runfor("example3Kernel1", N),
                        max_runfor("example3Kernel2", 13*12+2),
                        max_end());

    printf("Checking data read from FPGA.\n");

    printf("Expected: %f, returned: %f\n", expected, *data_out);
    max_close_device(device);
    max_destroy(maxfile);

    return 0;
}
```

Hvala na pažnji!

Saša Stojanović
stojsasa@etf.bg.ac.rs