

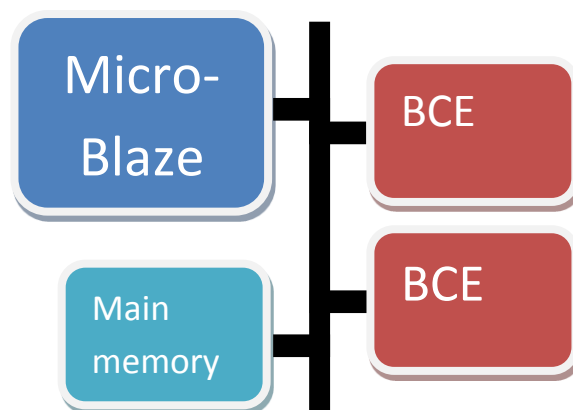


RAVAC – Compiler for Platforms with Programmable Accelerators

RAVAC is an easily extensible compiler for the ASVP/EdkDSP platform that allows much easier firmware generation. Also automatically schedules data transfers and allows usage of multiple accelerators at once.

ASVP Platform used in the Demonstrator

The ASVP/EdkDSP platform has a central processor and a set of programmable floating-point accelerators (Basic Computing Elements, BCEs).



Manual programming is time-consuming and error-prone, so in the SMECY project was development of the RAVAC compiler for this and similar platforms started.

The used ASVP platform has only two accelerators, but this allows it to fit to a low-cost FPGA Spartan 6.

ASVP Programming with RAVAC

The user specifies interesting loops in the source code using SMECY pragma vectorize. Such loops can be with advantage identified using BlueBee/QUAD developed at TU Delft.

An example of a simple loop can be seen here:

```
#pragma smecy vectorize(bce_sqr_sub, st_d2, inR, meansR,
    inG, meansG, inB, meansB)
for(int xk=0; xk < CONSTLEN*MOG_MODELS_PER_PIXEL; xk++)
{
    st_d2[xk] =
        (inR[xk] - meansR[xk])*(inR[xk] - meansR[xk]) +
        (inG[xk] - meansG[xk])*(inG[xk] - meansG[xk]) +
        (inB[xk] - meansB[xk])*(inB[xk] - meansB[xk]);
}
```

The loop is analyzed; a dataflow graph is constructed, and firmware for a BCE from this loop is generated. It consists of following operations (common subexpressions are optimized too):

```
DFU_VSUB
DFU_VSUB
DFU_VMUL
DFU_VSUB
DFU_VMAC
DFU_VMAC
```

Also instructions for starting this operation and to issue data transfers for the central processor are generated.

For any suitable loop that contains operations supported by BCEs can be firmware generated, another more complex example from the demonstrator application is shown here:

```
#pragma smecy vectorize(bce_caclhit_constlen,
    st_kHit_in, st_d2_k_in, models_k_varsum_in, k_val,
    models_k_weight_in, weight_indirect_in)
for(int x=0; x<CONSTLEN; x++)
{
    st_kHit_in[x] =
        (st_d2_k_in[x * MODELS_PER_PIXEL]
         < ((float) 6.25 * models_k_varsum_in[x*MODELS_PER_PIXEL]))
        ?
        ((st_kHit_in[x]<(float) 0.0)
         ?
         k_val[0]
         :
         ((weight_indirect_in[x]
           < models_k_weight_in[x * MODELS_PER_PIXEL])
          ?
          k_val[0]
          :
          st_kHit_in[x]))
        :
        st_kHit_in[x];
}
```

Another important feature of the RAVAC compiler is complete support of SMECY IR1 pragmas, specifically its OpenMP subset.

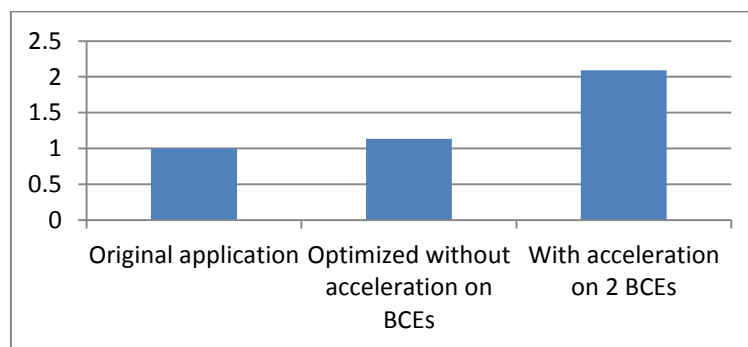
```
#pragma omp parallel for
for ( y=0; y<cdata->height; y++)
    mog_algorithm_line(...);
```

This allows running multiple operations at once and to overlap computations and communication.

Speedup Obtained using RAVAC Compiler for the FG/BG Demonstrator

For the FG/BG demonstrator, the application was first optimized, also by using BlueBee/QUAD tool, and partially rewritten to better fit the ASVP platform with hints from the . Using BlueBee/QUAD has shown to be very beneficial since it shows the main data transfers.

Speedups are reported for the ASVP platform two BCEs as shown in the first page. The main processor core does not use floating-point instructions.



Almost any suitable loop in the source code can be offloaded to be executed on a BCE. With a faster system bus and more accelerators, much higher speedup can be reached.

Advantages of the RAVAC Compiler

- Automatic BCE firmware generation.
- Created with the aim for extensibility when new BCE instructions are supported.
- The first compiler platform with runtime able to handle OpenMP pragmas for the MicroBlaze processor with PetaLinux operating system.
- Automatic usage of multiple accelerators at once.
- RAVAC runtime handles accelerators programming, data transfers, and computation starting.
- The same source code that is used for compilation can be executed on a host PC for much easier debugging and automatic testing.
- With RAVAC, it is possible to program the ASVP platform much faster than manually..

Contacts

For support, inquiries, or licensing options, please contact:

Adam Husár

Faculty of Information Technology, Brno University of Technology

Bozetechova 2

Brno, CZ

email: ihusar@fit.vutbr.cz, phone: +420 541 141 443