




Transforming Slow Software into Fast Hardware on a Zynq® All Programmable SoC using Vivado® HLS and SystemC/TLM-2.0

Presenter: **David C Black**
Senior Member of Technical Staff at Doulos

Example code & profile

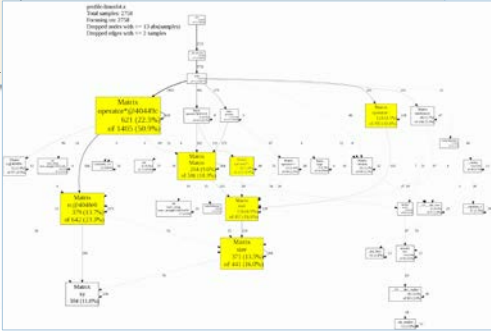


```
int main() {
#include "matrix.h"

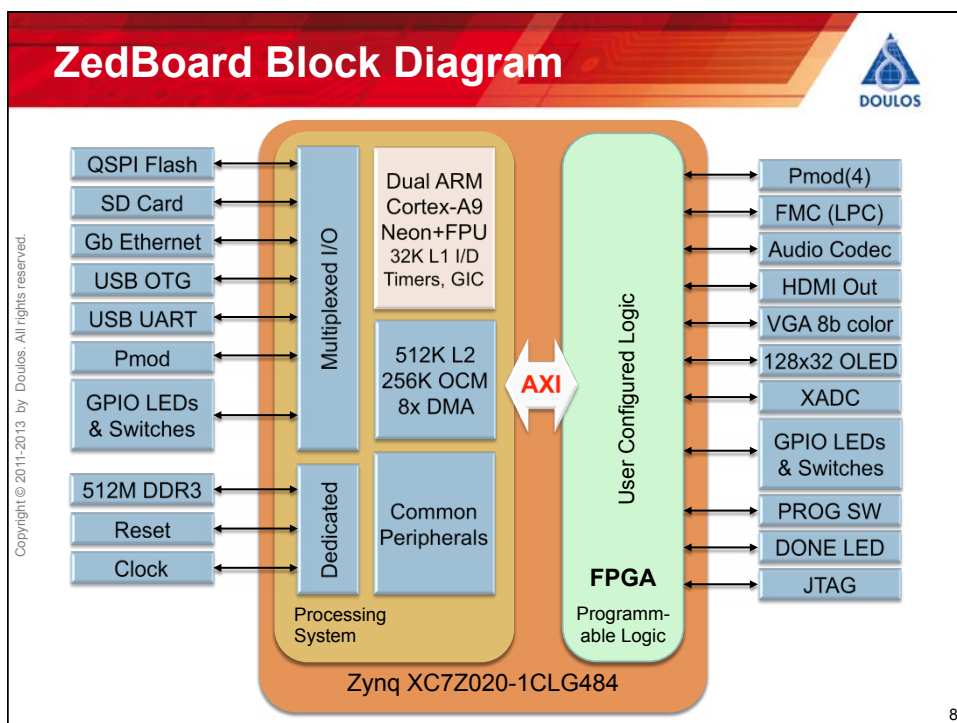
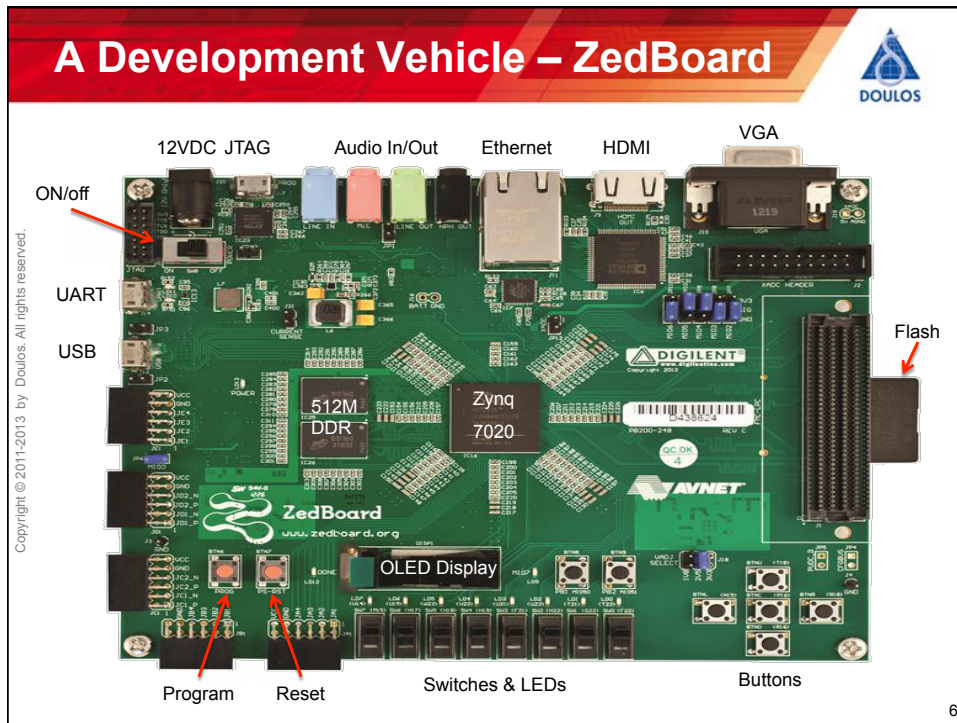
int main()
{
    Matrix lhs_matrix(3,4);
    Matrix rhs_matrix(4,3);
    Matrix result3_matrix(3,3);
    Matrix result4_matrix(4,4);
    lhs_matrix.setup_constants();

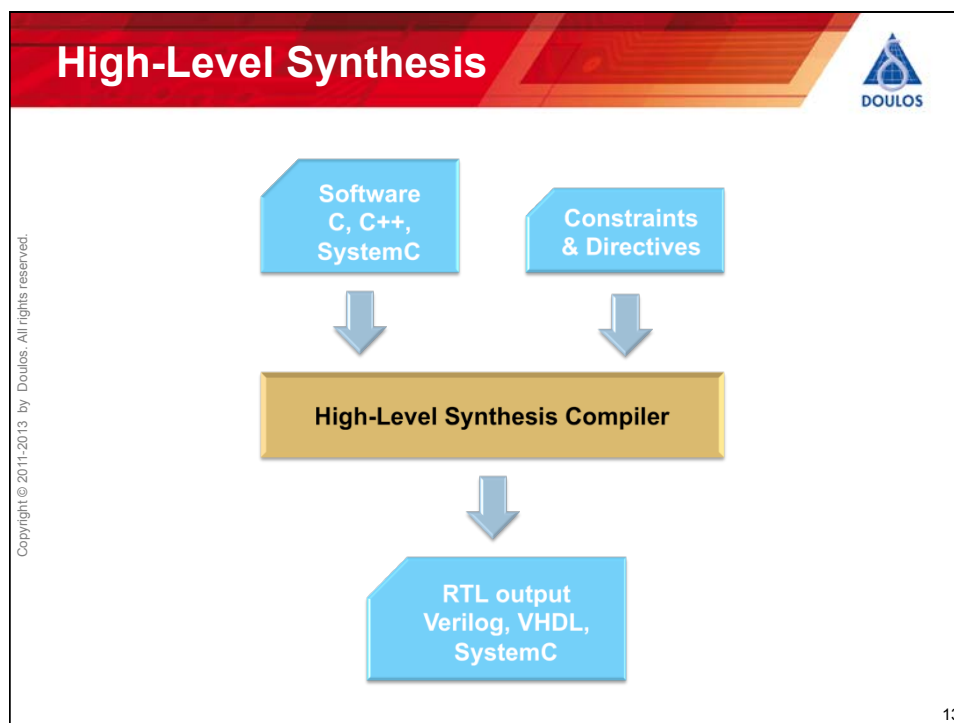
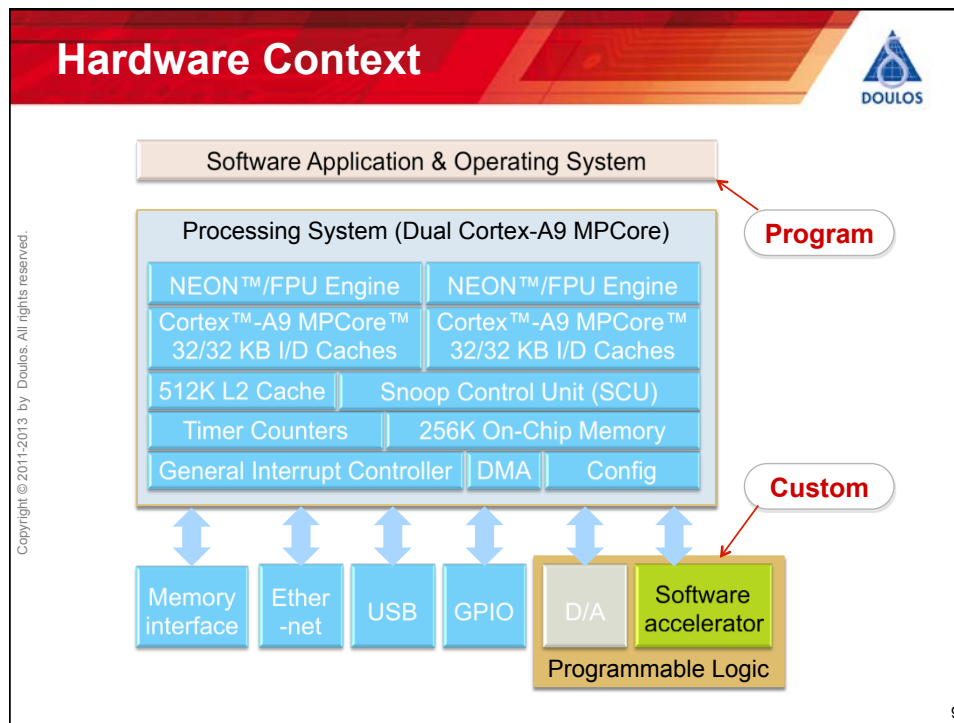
    // Do a bunch of operations
    for (int n=REPS; n!=0; --n)
    {
        rhs_matrix.read_input();
        result3_matrix
            = lhs_matrix * rhs_matrix
              + offset;
        result4_matrix
            = rhs_matrix * lhs_matrix
              * offset;
    }
    return 0;
}
```

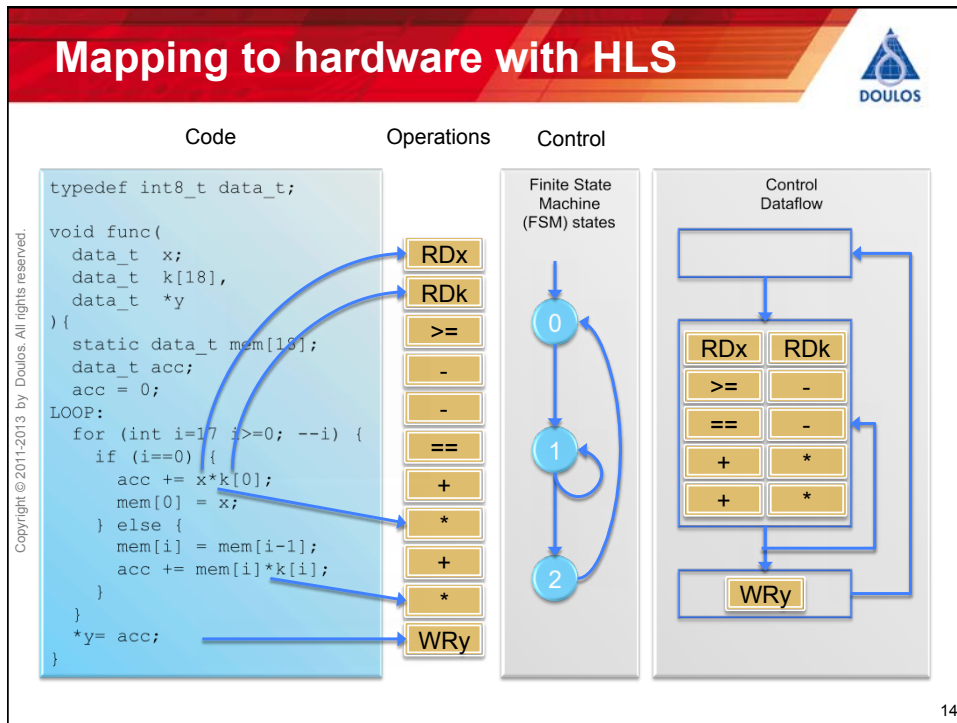
```
Matrix Matrix::operator*
(const Matrix& rhs) const {
    assert(m_cols == rhs.m_rows);
    Matrix result(m_rows, rhs.m_cols);
    for (int r=0; r!=m_rows; ++r) {
        for (int c=0; c!=rhs.m_cols; ++c) {
            int sum = 0;
            for (int i=0; i!=m_cols; ++i) {
                sum += rc(r,i)*rhs.rc(i,c);
            }
            result.rc(r,c) = sum;
        }
    }
}
```



Copyright © 2011-2013 by Doulos. All rights reserved.







Hardware Interface from Function

Copyright © 2011-2013 by Doulos. All rights reserved.

```

Data_t dev_hls( volatile Data_t reg[REGISTERS]
               , volatile Data_t mem[IMEM_SIZE])
{
    #pragma HLS interface ap_memory port=reg
    #pragma HLS interface ap_ctrl_hs port=return register
    #pragma HLS interface ap_memory port=mem
    #pragma HLS resource core=RAM_1P \
        metadata="-bus_bundle devreg" variable=reg
    #pragma HLS resource core=AXI4LiteS\
        metadata="-bus_bundle devreg" variable=return
    #pragma HLS resource core=RAM_1P \
        metadata="-bus_bundle devreg" variable=mem
    ...
    return retcode;
}

```

Interrupt

R0
R1
...
R15
Base
Command
Status

Local
RAM
16K x 32

16

Coding a matrix multiply

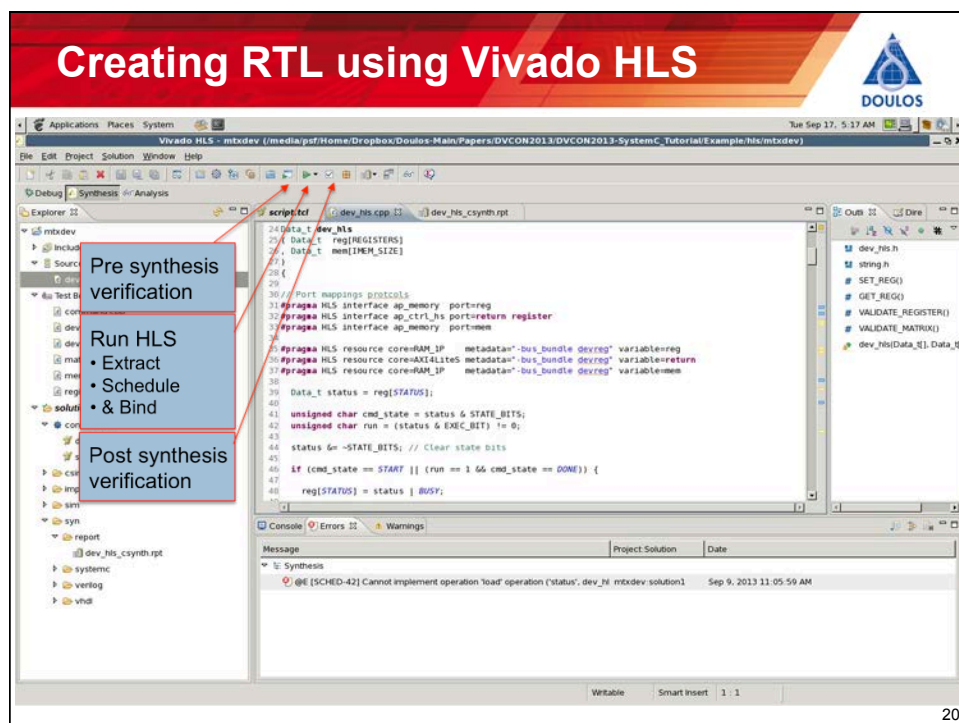
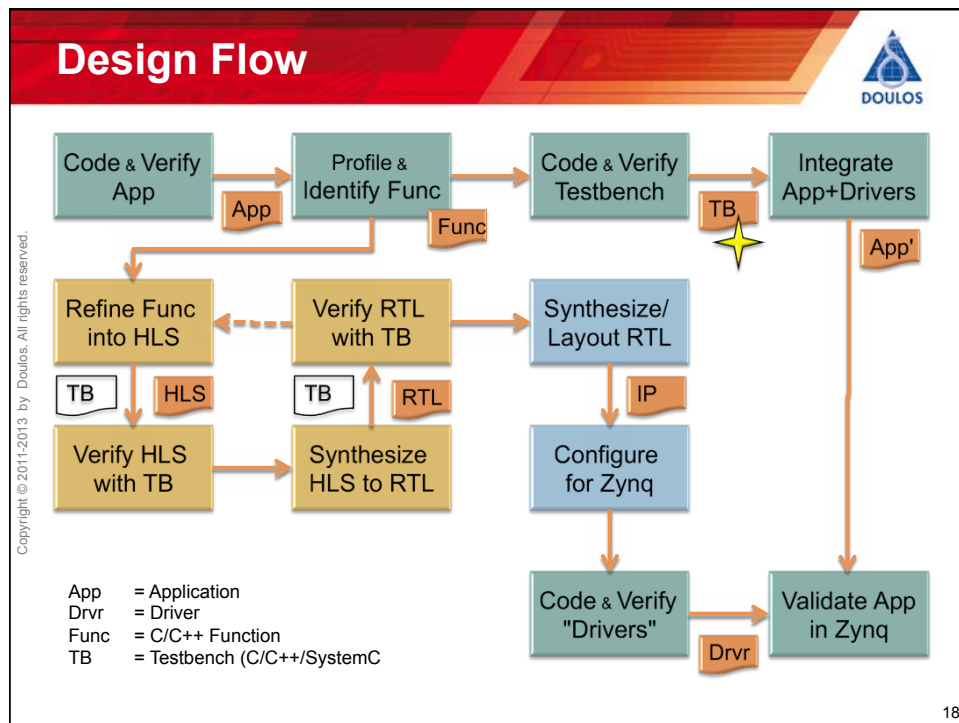
Copyright © 2011-2013 by Doulos. All rights reserved.

```

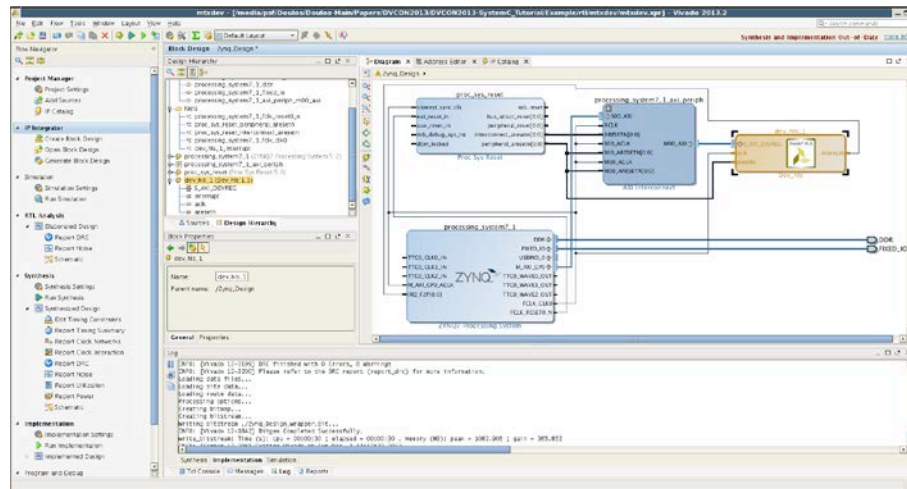
// dest = src1 x src2
Data_t dest_data, src1_data, src2_data;
for (Index_t r=0; r!= src1_rows; ++r) {
    for (Index_t c=0; c!= src2_cols; ++c) {
        dest_data = 0;
        for (Index_t i=0; i!= src1_cols; ++i) {
            src1_data = mem[ src1_ptr + src1_cols * r + i ];
            src2_data = mem[ src2_ptr + src2_cols * i + c ];
            dest_data += src1_data * src2_data;
        }
        mem[ dest_ptr + dest_cols*r + c ] = dest_data;
    }
}

```

17



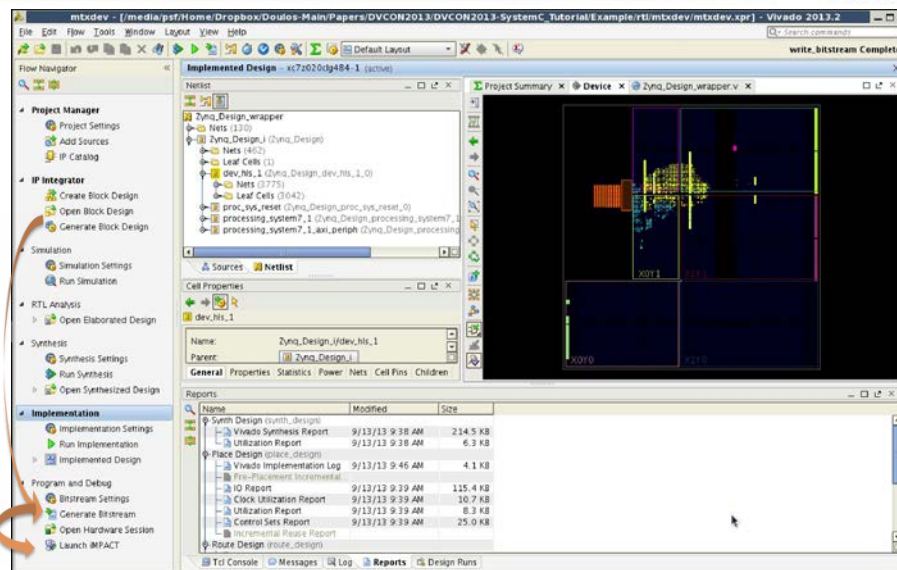
Integration using Platform Integrator



Copyright © 2011-2013 by Doulos. All rights reserved.

23

Creating Logic using Vivado



Copyright © 2011-2013 by Doulos. All rights reserved.

24

