



Universiteit Leiden
The Netherlands

Mapping Stream based Applications to an Intel IXP Network Processor using Compaan

Sjoerd Meijer (PhD Student)
University Leiden, LIACS
smeijer@liacs.nl

Outline

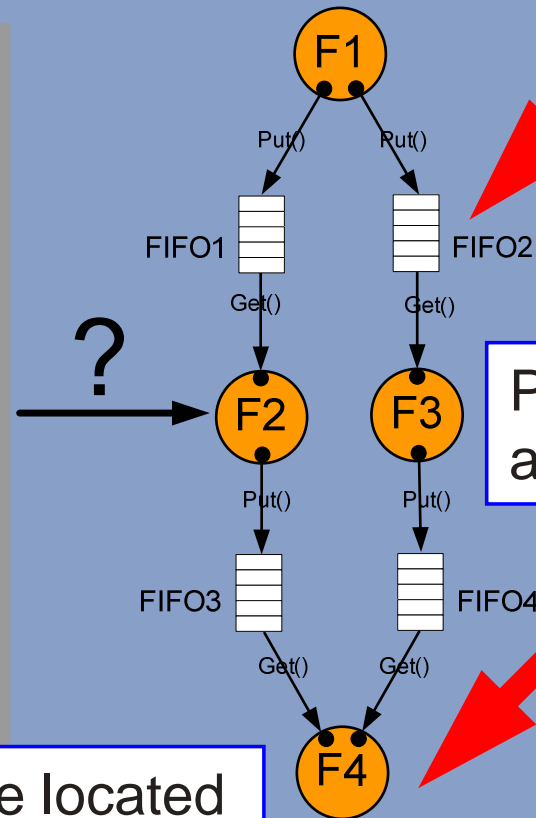
- Need for multi-processor platforms
- Problem: how to program them
- Solution: compiler support
- Realization: the IMCA back-end
- Conclusion

Problem Description

Sequentially Ordered

```
for( int t=1; t<=P; t++){
  for( int i=1; i<=M; i++ ){
    for( int j=4; j<=N; j++ ){
      r1[i+1][j-3] = F1(...);
    }
  }
  for( int l=3; l<=M; l++ ){
    for( int m=3; m<=N-1; m++ ){
      if ( l+m<= 7 ){
        r2[l][m] = F2( r1[l-1][m-2] );
      }
      if ( l+m>=8 ){
        r2[l][m] = F3( r1[l][N-3] );
      }
      ... = F4( r2[l][m] );
    }
  }
}
```

Storage arrays (R1) are located in Global Memory

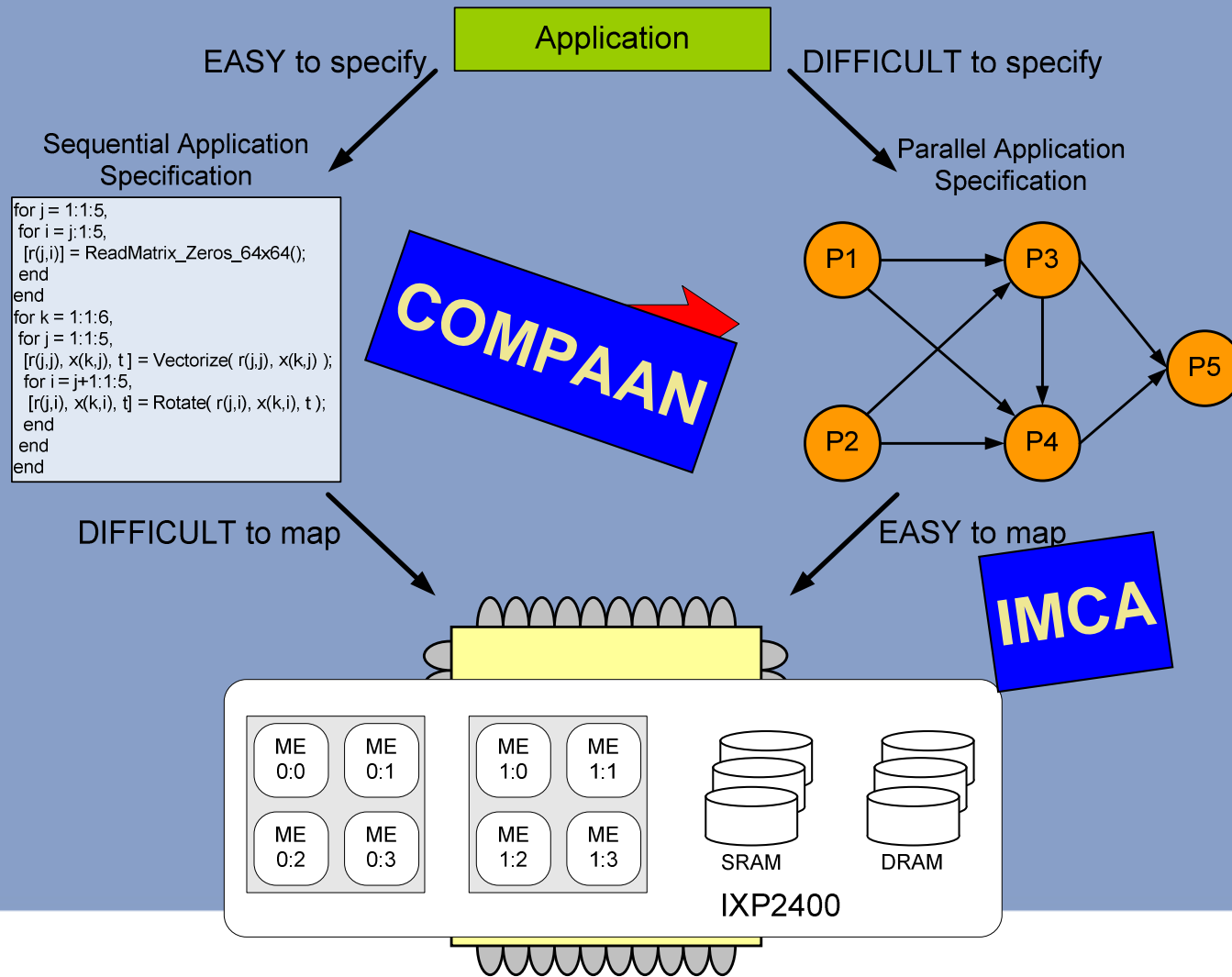


Communicate via unbounded FIFOs

Processes run autonomously

We need tools to convert the sequential program to a parallel equivalent

Our Solution



Questions

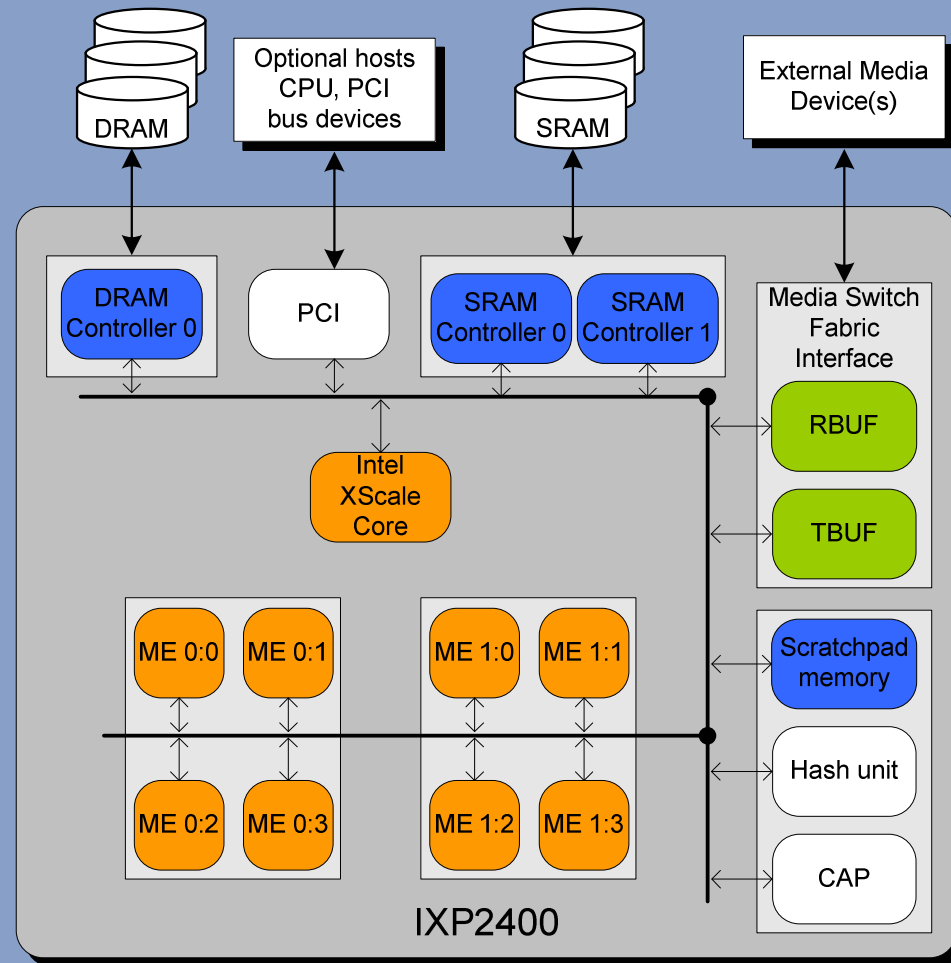
- Q1: Can we use the IXP architecture for stream-based applications?
- Q2: Can we map applications written as a KPN onto the IXP?
- Q3: Can we program the IXP using Compaan?

Intel IXP2400 Network Processor

- Optimized for streaming
 - 2.6 Gbit ethernet connection
- Build to operate in real-time on internet traffic
 - 8 microengines with 8 hardware supported threads
- Completely programmable in C
 - A SDK available for programming

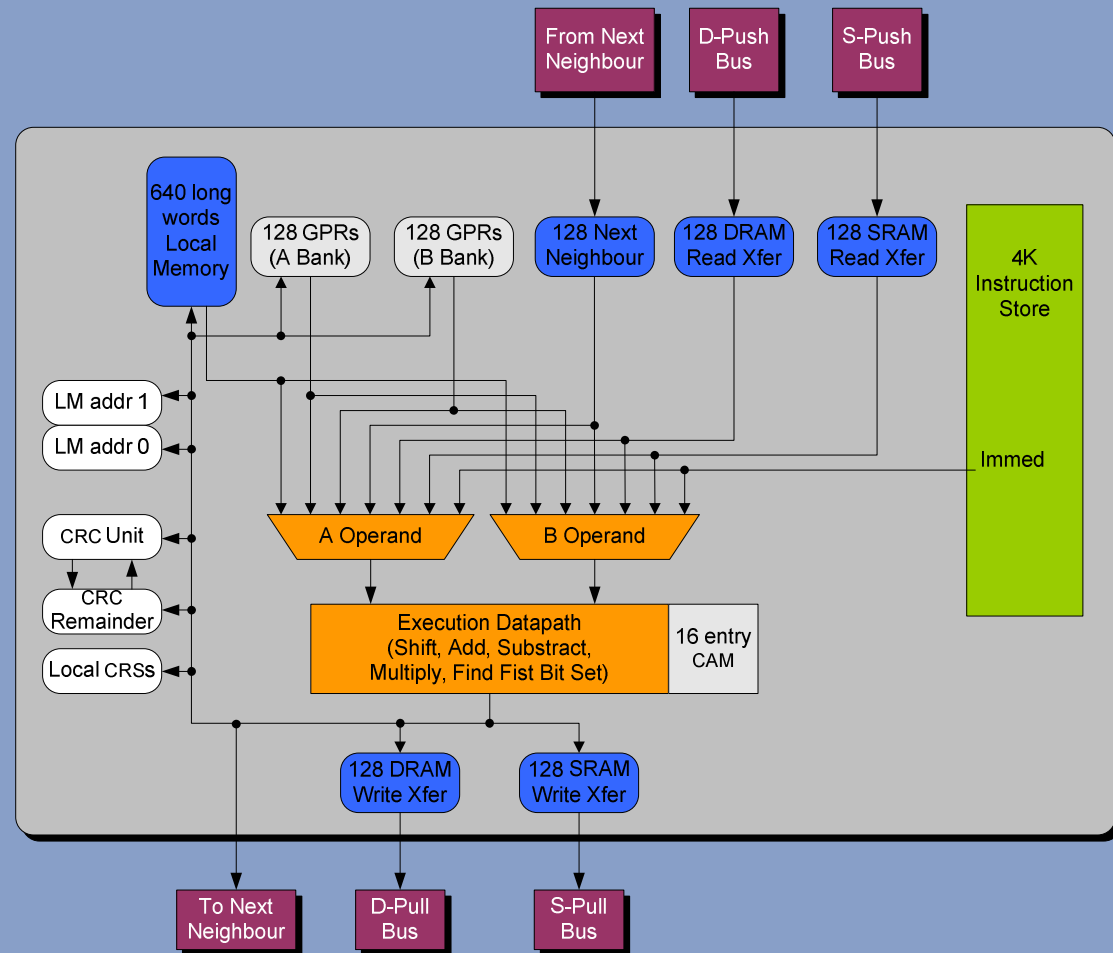
IXP2400

- 8 microengines
- 1 XScale core
- MSF interface
- SRAM: 128 MB
- DRAM: 1 GB
- Scratchpad: 4 K



Microengine

- Local memory: 640 K
- Registers
 - GPR
 - Read Xfer
 - Write Xfer
 - Next neighbour
- Instruction store: 4 K
- 8 Threads



Intel IXP Network Processor

- Not used on a large scale
- Difficult to program:
 - Write code per microengine
 - Infrastructure
 - Synchronization
 - Non-unified complex memory model

Conclusion: easier programming needed

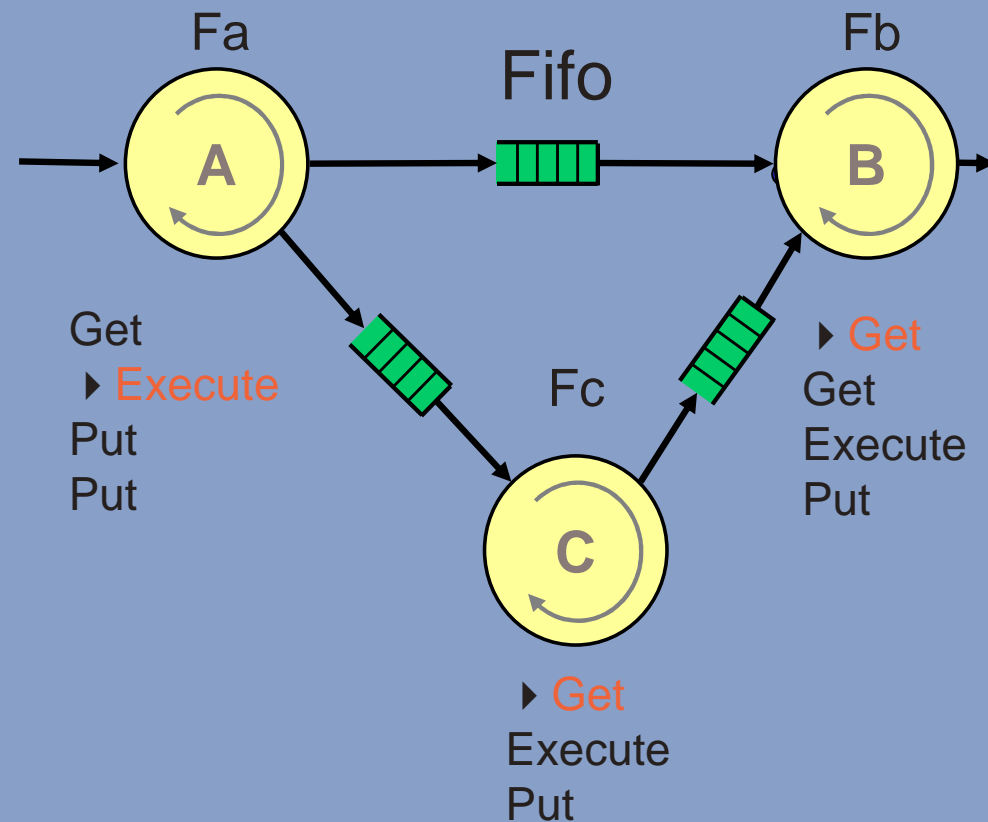
Static Affine Nested Loop Programs

- Nested loop: all statements occur within loops
- Static: control flow known at compile time
- Affine: $ax+b$ expressions
- Explicit array references are required to extract data-level parallelism in the application

```
01 int i,j;
02 matrix A;
03 for (i = 1, i < 5, i++) {
04     for (j = i, j < 12, j++) {
05         A[i,j] = 3*(i+j) - 3;
06         if (j > 3)
07             A[i,j+1] = i+j;
08     }
09 }
```

Kahn Process Network (KPN)

- Process Networks
 - Processes run autonomously
 - Communicate via unbounded FIFOs
 - Synchronize via blocking read
- Process is either
 - executing (Execute)
 - communicating (Put/Get)
- **BENEFITS:**
 - Deterministic Behavior
 - Distributed Control
 - Distributed Memory



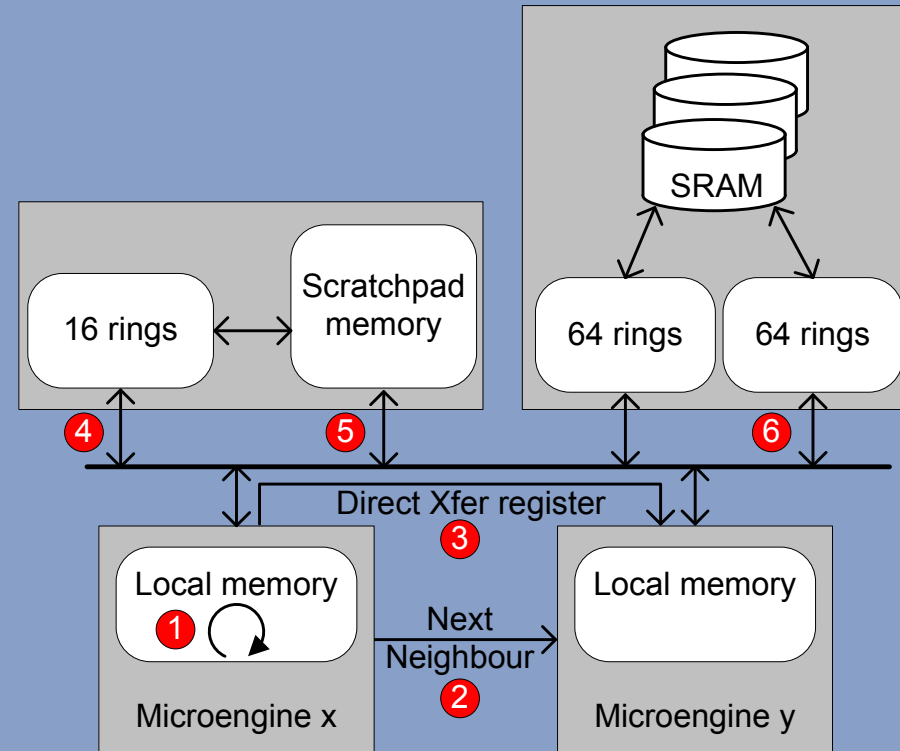
Signals

- Synchronization between IXP elements
 - Between threads and microengines
 - Memory access
- Wait for signal → context switch

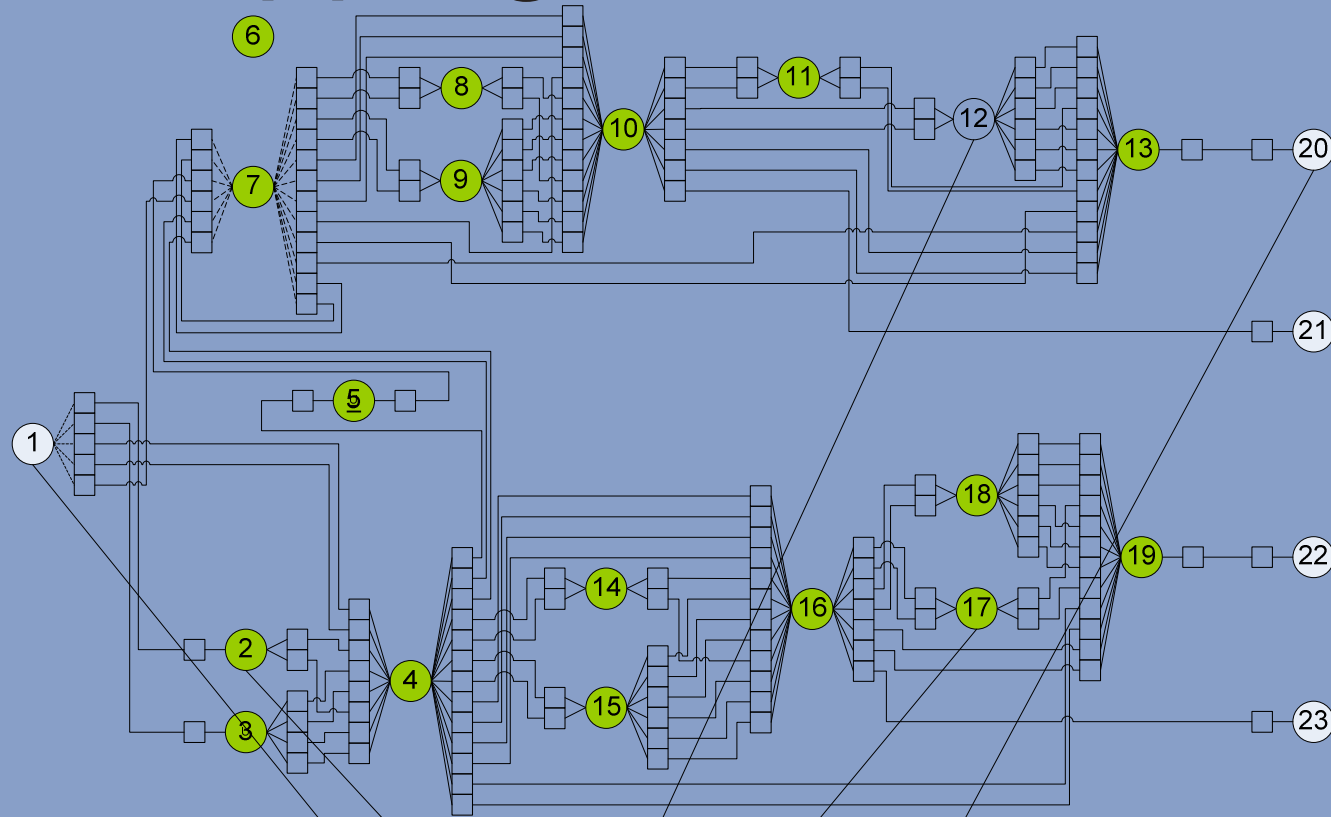
```
__declspec(sram_read_reg) x; SIGNAL sig;  
__declspec(scratch)* addr = 0x400;  
scratch_read(&x, addr, 1, sig_done, sig);  
do_other_work();  
__wait_for_all(&sig);  
y = x;
```

FIFO mappings

- Small and frequently used:
 - Scratchpad HW
 - (Next neighbour rings)
- Small and less frequently used:
 - Scratchpad SW
- Large and/or not frequently used:
 - SRAM rings



Process mapping

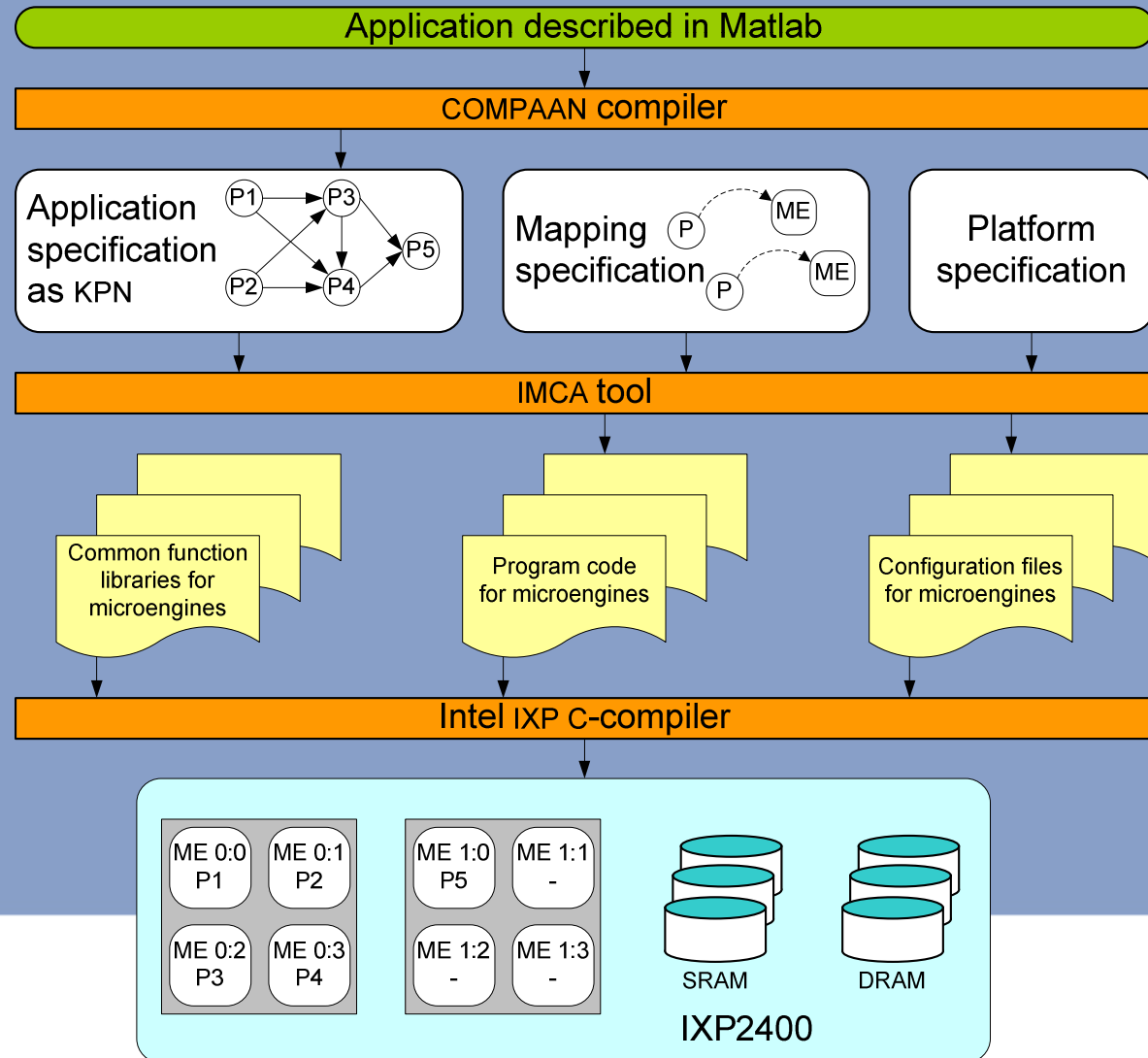


Thread	Microengine							
	0:0	0:1	0:2	0:3	1:0	1:1	1:2	1:3
1			1	2	3	4	5	
2			6	7	8	9	10	
3			11	12	13	14	15	
4			16	17	18	19	20	
5			21	22	23			

Tool Flow Overview

- IMCA:

IXP
Mapper for
Compaan
Applications



Code generation

- Visitor design pattern
 - Visits platform description, writes code per element
- One “C” file per microengine
- FIFOs accessed uniformly
 - Static functions to implement FIFO code
 - *Port* struct to specify FIFO variables

Results

- QR algorithm
 - 5 nodes
 - 12 FIFOs

Arch.	# clock cycles	CPU freq.	Time micro secs.
IXP	40247	600 Mhz	67
FPGA, 5 MB	3865	100 Mhz	39
FPGA, full HW	213	108 Mhz	2

Discussion

- Work presents a first try, still many open issues
 - Selection of right communication channel
 - Binding of the KPN processes to threads and microengines
 - MSF takes a lot of resources, what is the minimum required.
- Future of the IXP is uncertain, perhaps the Cell Processor is an interesting next research platform

Conclusion

- Q1: The IXP can be used for streaming applications
 - Yes, showed it for QR and DWT
- Q2: We automatically mapped QR
 - Yes, we can map the FIFO communication onto the communication channels and the processes on the threads
- Q3: IMCA back-end generates IXP code
 - Yes, we can use Compaan to automatically generate the Processes and FIFO channels that are subsequently mapped on the IXP