

# Modeling Adaptive Streaming Applications with Parameterized Polyhedral Process Networks

**Teddy Zhai, Hristo Nikolov, Todor Stefanov**

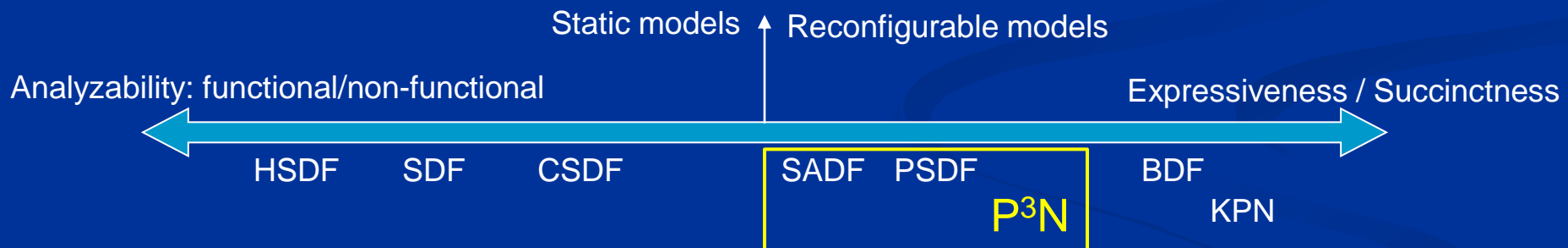
Leiden Embedded Research Center,  
Leiden Institute of Advanced Computer Science  
Leiden University, The Netherlands

*tzhai@liacs.nl*



# Background

- Emerging of adaptive streaming applications
  - Change of modes
  - Change of processing rates
- Adaptation of *Models-of-Computation* in the Electronic System-level design (ESL) methodology
  - Decidable dataflow models ( (H)SDF, CSDF, *etc.*)
  - Undecidable dataflow models ( BDF, KPN, *etc.* )



(H)SDF:	(Homogeneous)Synchronous Dataflow
CSDF:	Cyclo-Static Dataflow
SADF:	Scenario-Aware Dataflow
PSDF:	Parameterized SDF
BDF:	Boolean-controlled Dataflow
<b>P<sup>3</sup>N:</b>	<b>Parameterized Polyhedral Process Networks</b>
KPN:	Kahn Process Network

# Related Work

- Hierarchical reconfiguration of dataflow models

[ S. Neuendorfer and E. Lee: Hierarchical Reconfiguration of Dataflow Models, *MEMOCODE*, 2004 ]

How to extract dependence relation between parameters is not discussed

- Parameterized Synchronous Dataflow (PSDF)

[ B. Bhattacharyya and S. Bhattacharyya : Parameterized dataflow modeling for DSP systems, *IEEE Trans. Sig. Proc.*, 2001 ]

Large run-time overhead

- Scenario Aware Dataflow (SADF)

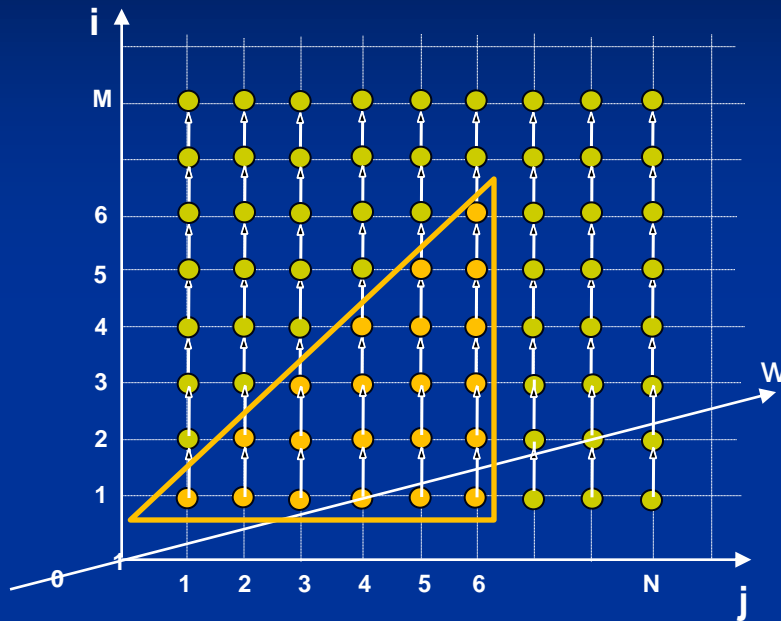
[ B. Theelen *et al.* : A scenario-aware data flow model for combined long-run average and worst-case performance analysis, *MEMOCODE*, 2006 ]

Limited expressiveness

Incurred performance overhead is not evaluated on MPSoC platforms

# Polytope Model

- Parametric polyhedron (unbounded)



$$\begin{aligned} 1 &\leq w, \\ 1 &\leq j \leq N, \\ 1 &\leq i \leq M, \\ &i \leq j \end{aligned}$$

```
While ( 1 ) {
  for ( j = 1; j <= N; j++ ) {
    for ( i = 1; i <= M; i++ ) {
      if ( i <= j ) {
        ...
      }
    }
  }
}
```

- Polytope (bounded polyhedron)

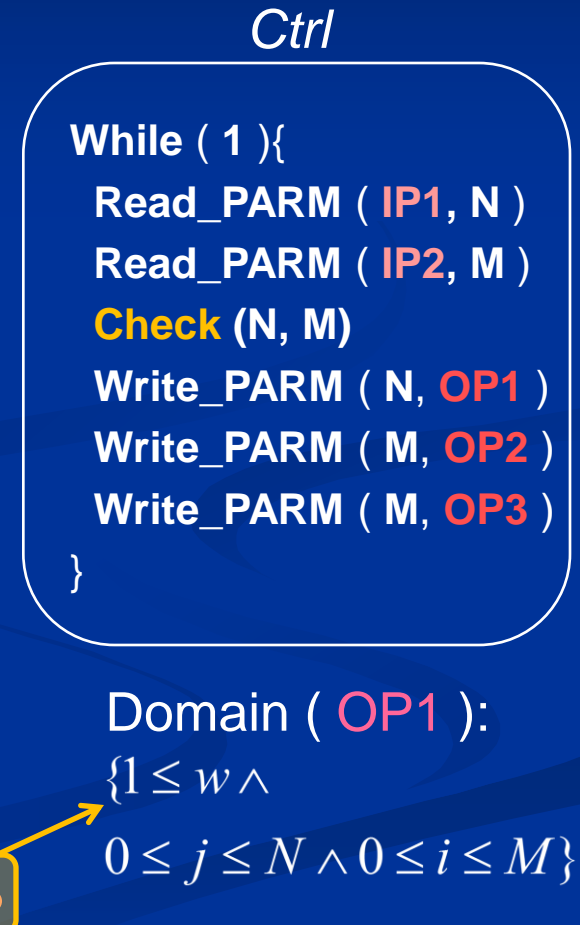
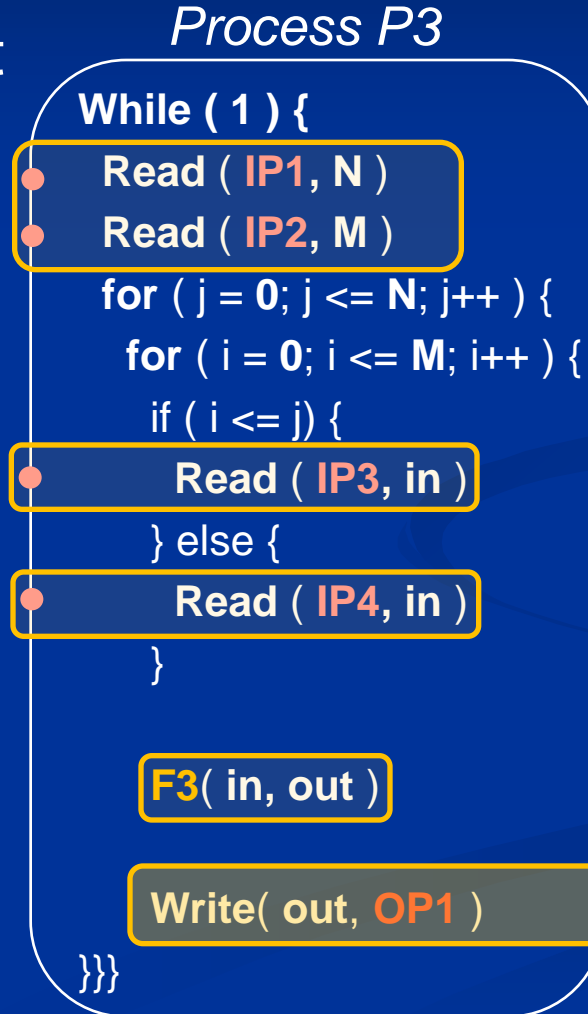
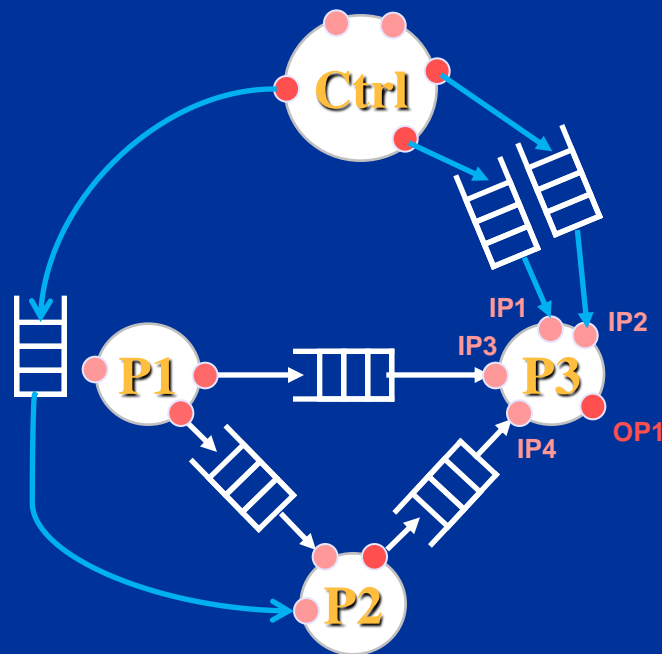
$$\{w=1 \wedge 1 \leq j \leq 6 \wedge 1 \leq i \leq 6 \wedge i \leq j\}$$

- Lexicographically ordered iteration points

$$\begin{pmatrix} w \\ j \\ i \end{pmatrix} : \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix} \prec \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$$

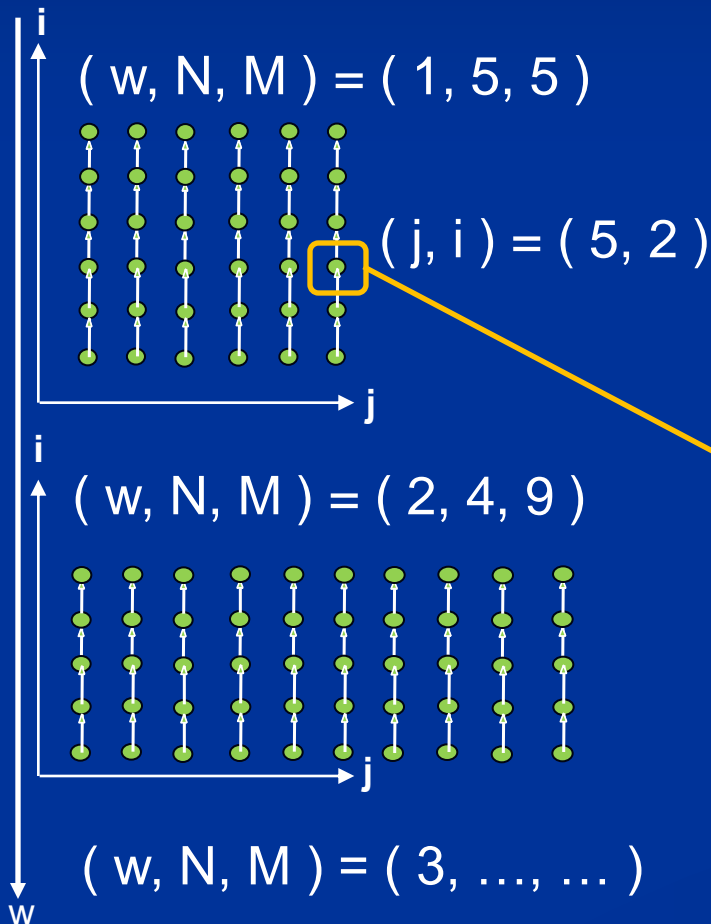
# Definition

- Parameterized Polyhedral Process Network ( $P^3N$ )
- Dataflow / Control process
- Input / Output port
- Function
- Domain



# Operational Semantics

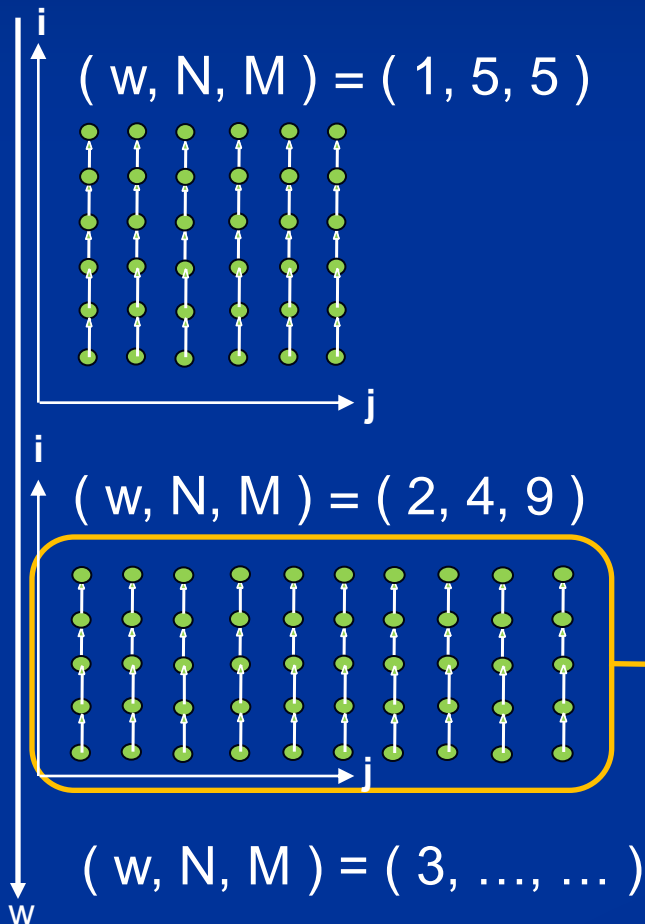
- Process iteration: READ-EXEC-WRITE sections
- Process cycle: an ordered set of process iterations (polytope)



```
While ( 1 ) {  
  Read ( IP1, N )  
  Read ( IP2, M )  
  for ( j = 0; j <= N; j++ ) {  
    for ( i = 0; i <= M; i++ ) {  
      if ( i <= j ) {  
        Read ( IP3, in )  
      } else {  
        Read ( IP4, in )  
      }  
  
      F3( in, out )  
  
      Write( out, OP1 )  
    }  
  }  
}
```

# Operational Semantics

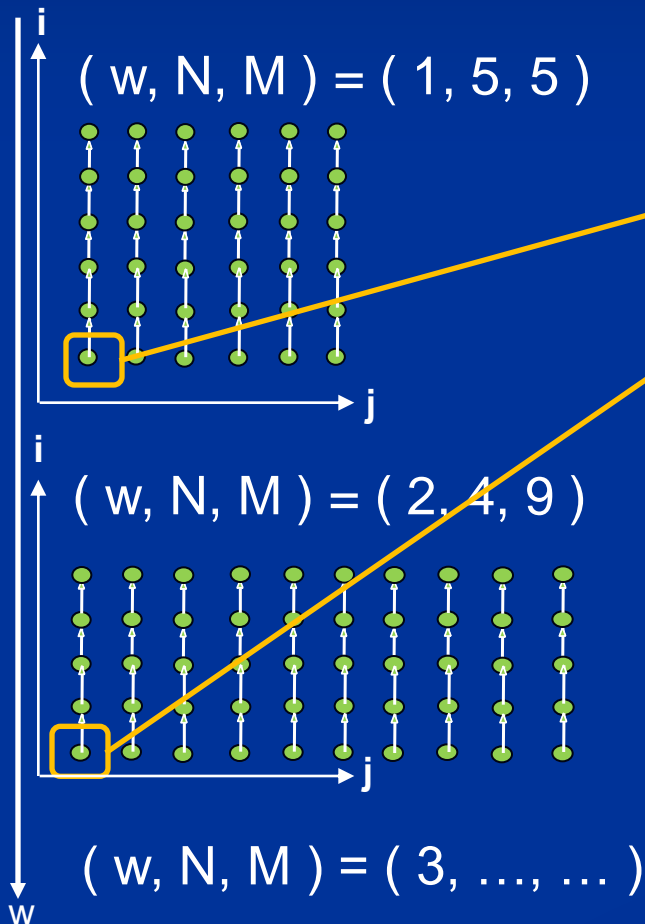
- Process iteration: READ-EXEC-WRITE sections
- Process cycle: an ordered set of process iterations (polytope)



```
While ( 1 ) {  
  Read ( IP1, N )  
  Read ( IP2, M )  
  for ( j = 0; j <= N; j++ ) {  
    for ( i = 0; i <= M; i++ ) {  
      if ( i <= j ) {  
        Read ( IP3, in )  
      } else {  
        Read ( IP4, in )  
      }  
    }  
    F3( in, out )  
    Write( out, OP1 )  
  }  
}
```

# Operational Semantics

- Process iteration: READ-EXEC-WRITE sections
- Process cycle: an ordered set of process iterations (polytope)



```
While ( 1 ) {
```

```
  Read ( IP1, N )
```

```
  Read ( IP2, M )
```

```
  for ( j = 0; j <= N; j++ ) {
```

```
    for ( i = 0; i <= M; i++ ) {
```

```
      if ( i <= j ) {
```

```
        Read ( IP3, in )
```

```
      } else {
```

```
        Read ( IP4, in )
```

```
      }
```

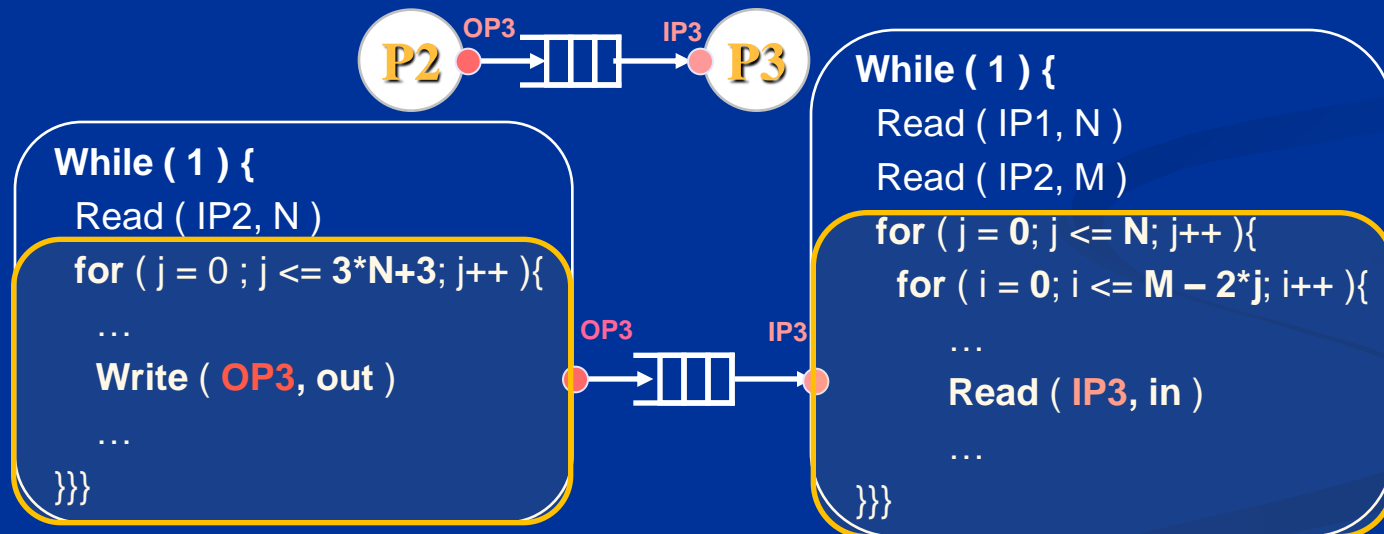
```
      F3( in, out )
```

```
      Write( out, OP1 )
```

```
  } }
```

# Consistency of P<sup>3</sup>N

- Consistency of the P<sup>3</sup>N model ensures:
  - Deadlock-free execution
  - No buffer overflow
- Consistency of P<sup>3</sup>N: #Tokens written to any channel is equal to the #Tokens read from it after one corresponding process cycle

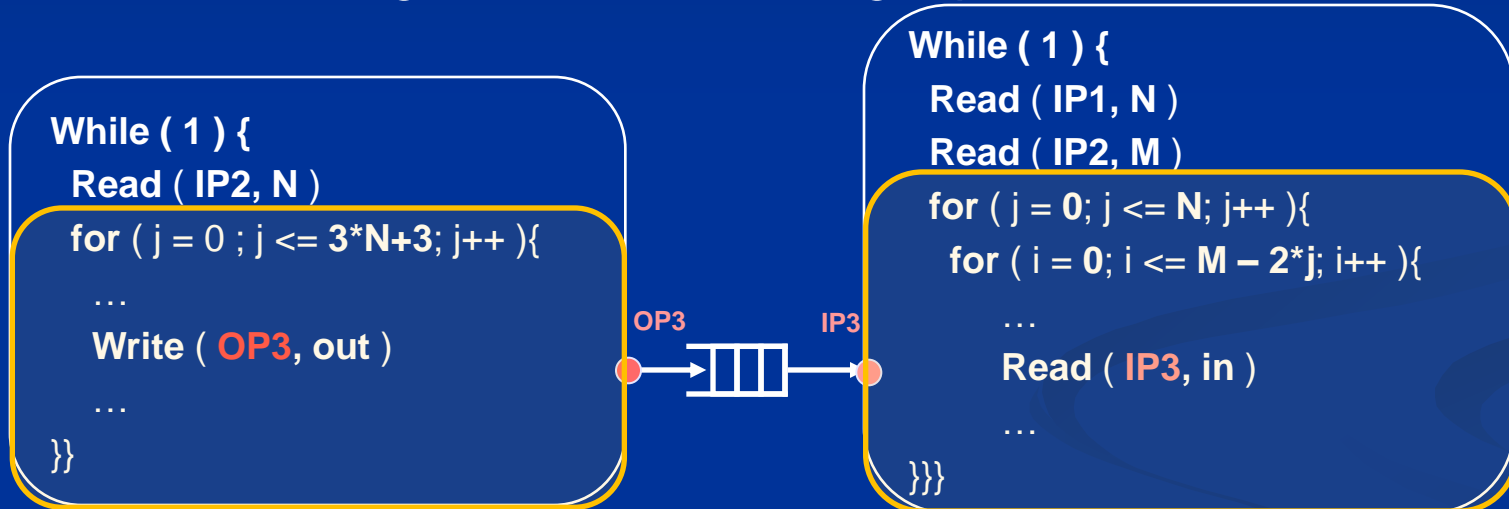


( N, M ) = ( 7, 8 ) ?  
 ( N, M ) = ( 7, 3 ) ?

Which combinations (N, M) do ensure the consistency ?

# Consistency Check

- The consistency condition has to be checked at run-time
  - Parameter values are unknown at design-time
- Our design-time analysis
  - Counting the number of integer points in domains: *Barvinok* library



Domain (**OP3**):  $\{0 \leq j \leq 3N + 1\}$   
 $|OP3| = 3N + 4$

Domain (**IP3**):  $\{0 \leq j \leq N \wedge 0 \leq i \leq M - 2j\}$

$(N, M) = (7, 8)$

$$3 \times 7 + 4 = 1 + \frac{3}{4} \times 8 + \frac{1}{4} \times 8^2 + \frac{1}{4} \times 8 - \frac{1}{4} \times 0$$

$$25 = 25$$

$$|IP3| = \begin{cases} 1 + M + NM - N^2 & \text{if } C1 \\ 1 + \frac{3}{4}M + \frac{1}{4}M^2 + \frac{1}{4}M - \frac{1}{4}\{0,1\}_M & \text{if } C2 \end{cases}$$

$C1 = \{2N \leq M - 1\}$   
 $C2 = \{2N \geq M\}$

# Consistency Analysis

- For each channel (OP, IP), subtraction between (quasi)polynomials  $|OP| - |IP|$  yields
  - **Case 1:** result = 0, all parameter values satisfying condition C preserve the consistency condition
  - **Case 2:** result = *non zero constant*, no parameter values satisfying condition C preserve the consistency condition
  - **Case 3:** result = a (quasi-)polynomial, only a part of parameter values satisfying condition C preserve the consistency condition
- Function *check* in the control process

```
Check ( M, N){
```

```
  tab = {
```

```
    ( 6, 4 ),
```

```
    ( 8, 7 ),
```

```
    ( 15, 12 ),
```

```
  }
```

```
  ...
```

```
}
```

```
Check ( M, N){
```

```
  // condition C1
```

```
  if ( 2N <= M - 1 )
```

```
    res = M + N×M - N2 - 3N - 3
```

```
  // condition C2
```

```
  if ( 2N >= M )
```

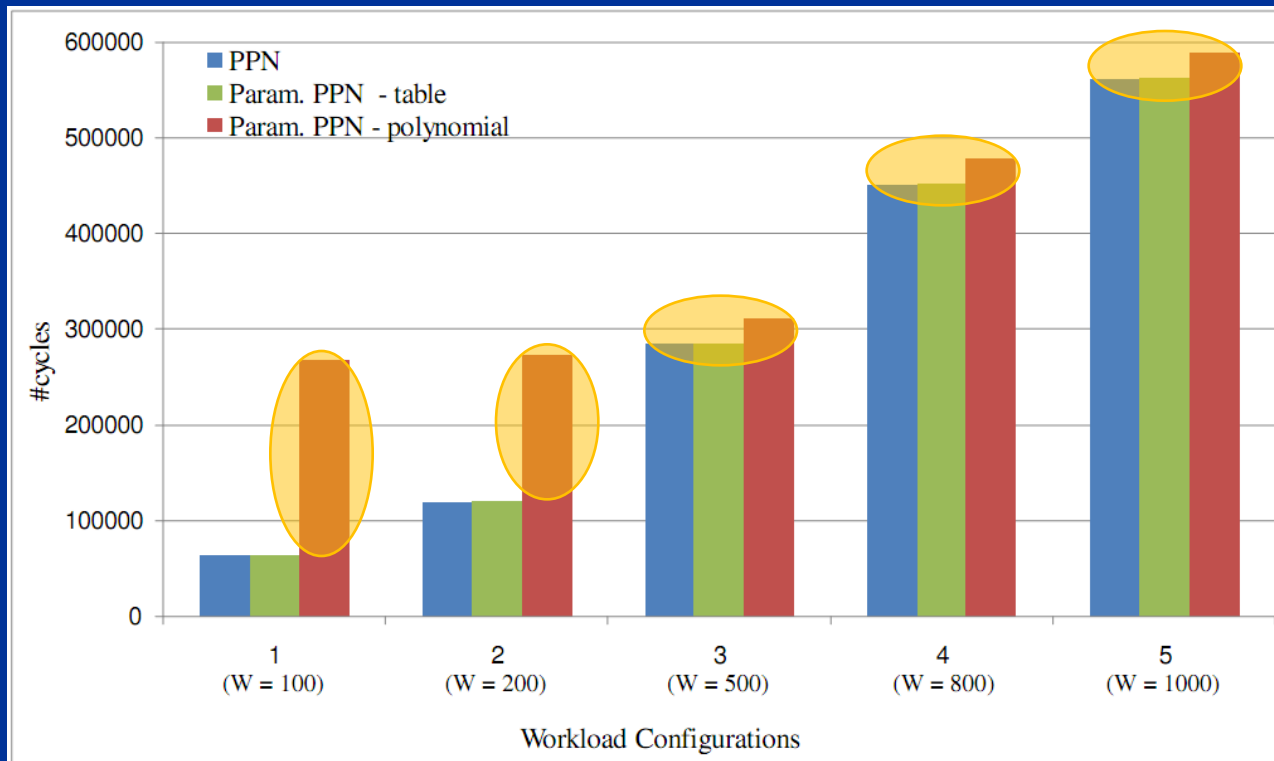
```
    res = 3M/4 + M2/4 + M/4 - {0,1}M/4 - 3N - 3
```

```
  ...
```

```
}
```

# Experimental Results

- Prototyping a P<sup>3</sup>N on a Virtex2 FPGA
- The control process evaluates complex polynomials at run-time



# Conclusion

- A new **model**, called Parameterized Polyhedral Process Networks ( $P^3N$ ), to capture adaptive application behavior
- A design-time approach to extract dependence relations between run-time parameters
- **Consistent execution** of the  $P^3N$  model is preserved at run-time
- Reconfiguration overhead is quantified on an FPGA-based MPSoC platform

# Thanks for your attention

*Modeling Adaptive Streaming Applications with  
Parameterized Polyhedral Process Networks*

Teddy Zhai  
Leiden Embedded Research Center  
Leiden Institute of Advanced Computer Science  
Universiteit Leiden  
Niels Bohrweg 1  
2333 CA Leiden, The Netherlands

Email: [tzhai@liacs.nl](mailto:tzhai@liacs.nl)  
URL: <http://www.liacs.nl/~tzhai>