Operation Scheduling, Binding and Data Routing for Run-Time Reconfigurable Architecture

E. Raffin - C. Wolinski – F. Charot – **K. Kuchcinski** – S. Guyetant – S. Chevobbe – E. Casseau

technicolor   UNIVERSITÉ DE RENNES 1   INRIA   LUNDS UNIVERSITET   cea list

Outline

The ROMA reconfigurable processor

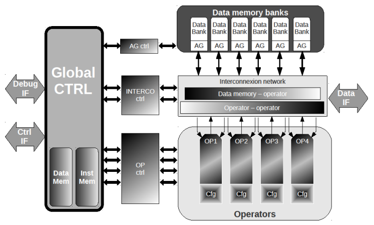
Coarse Grain Reconfigurable Architecture

✓ Main features

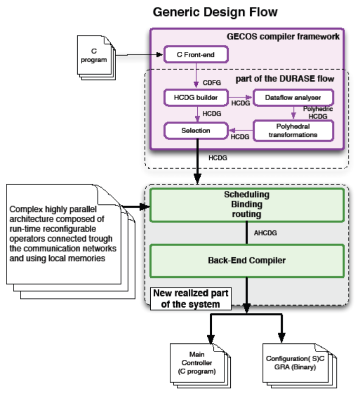
- Two **reconfigurable interconnection networks**
- Up to 8 **complex low power reconfigurable operators**
- Up to 14 **local memories**
- Reconfiguration in **1 cycle**
- **Multimedia** dedicated address generators and operators

✓ Physical features

- 250 MHz (TSMC 90nm)
- 1mm²



Our contribution in a simplified ROMA tool chain


Generic Design Flow
C program
GECOS compiler framework
C Front-end
CDFG
part of the DURASE flow
HCDG builder
HCDG
Dataflow analyser
Polyhedric HCDG
Selection
Polyhedral transformations
Complex highly parallel architecture composed of run-time reconfigurable operators connected trough the communication networks and using local memories
Scheduling Binding routing
AHCDG
Back-End Compiler
New realized part of the system
Main Controller (C program)
Configuration(S)C GRA (Binary)

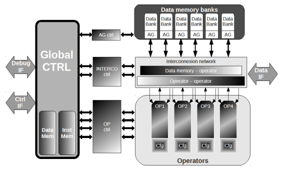
Architecture model

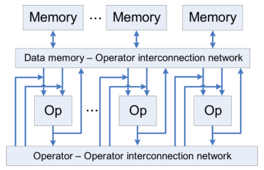
Goal: Model the ROMA architecture in a generic and parameterized way

➢ Memories:
- Number and size (one data port)
- Read/Write operation latencies (constant)
- Each memory is identified by a unique ID

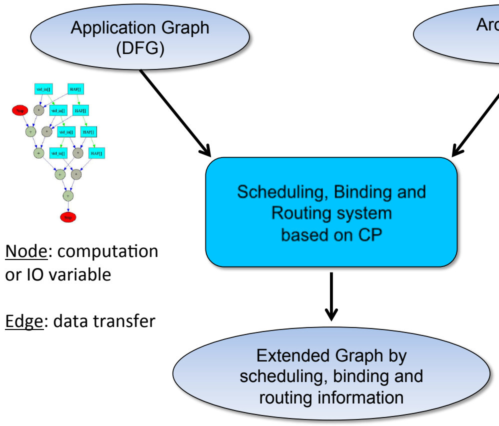
➢ Operators:
- Each operator is identified by a unique ID
- Operators can be heterogeneous
- Each Operator has 2 inputs and 1 output port

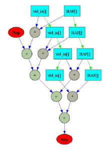
➢ Interconnection networks:
- The memory-operator network is a full Xbar
- The operator-operator network topology is parameterized by a connection matrix
- Network latencies

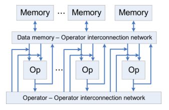



Memory
Memory
Memory
Data memory – Operator interconnection network
Op
Op
Op
Operator – Operator interconnection network

Constraint model for scheduling, binding and routing


Application Graph (DFG)
Architecture model
Scheduling, Binding and Routing system based on CP
Extended Graph by scheduling, binding and routing information




Memory
Memory
Memory
Data memory – Operator interconnection network
Op
Op
Op
Operator – Operator interconnection network

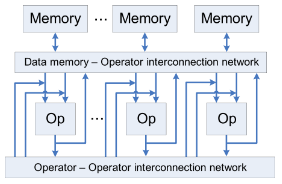
Node: computation or IO variable

Edge: data transfer

Constraint model for scheduling, binding and routing

➢ Communication constraints

- Which network is used for a data transfer?
- How to constrain a network use by its latency and its topology?


Memory
Memory
Memory
Data memory – Operator interconnection network
Op
Op
Op
Operator – Operator interconnection network

Constraint model for scheduling, binding and routing

➢ Timing constraints

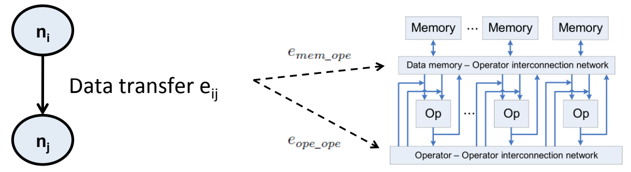
- How to ensure the precedence constraints of the application graph?
- How to adapt precedence constraints according to the network used?

Constraint model for scheduling, binding and routing

- Resource sharing constraints
  - How to ensure that memory and computation resources are not used at the same time?
  - How to ensure that we do not exceed the memory size limit?

Communication constraints

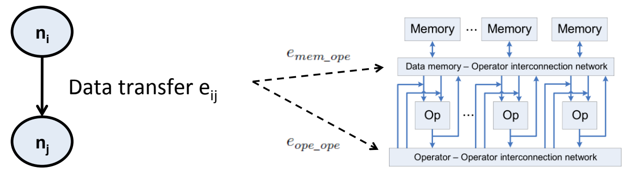
- Network choice



n_i
n_j
Data transfer e_{ij}
e_{mem_ope}
e_{ope_ope}
Memory
Memory
Memory
Data memory – Operator interconnection network
Op
Op
Op
Operator – Operator interconnection network

Communication constraints

- Network choice



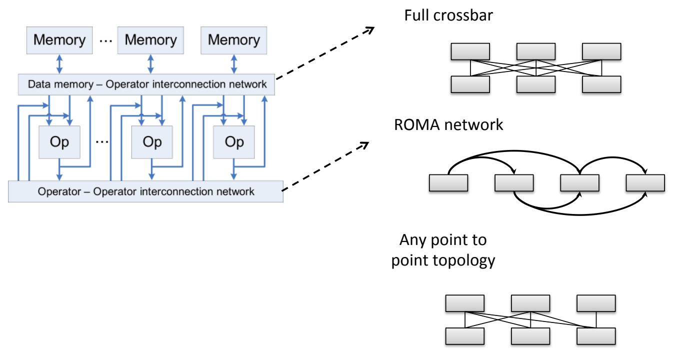
n_i
n_j
Data transfer e_{ij}
e_{mem_ope}
e_{ope_ope}
Memory
Memory
Memory
Data memory – Operator interconnection network
Op
Op
Op
Operator – Operator interconnection network

$$\forall e \in E : e_{mem\_ope} + e_{ope\_ope} = 1$$

$$\forall e_{ij} = (n_i, n_j) \in E | n_i \in IOs \wedge n_j \in OPs \vee$$
$$n_i \in OPs \wedge n_j \in IOs :$$
$$e_{i,j_{mem\_ope}} = 1$$

Communication constraints

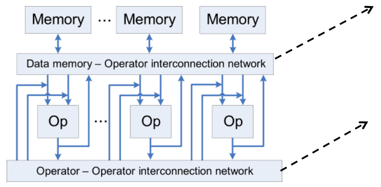
- Network topology



Memory
Memory
Memory
Data memory – Operator interconnection network
Op
Op
Op
Operator – Operator interconnection network
Full crossbar
ROMA network
Any point to point topology

Communication constraints

➢ Network topology


Memory
…
Memory
Memory
Data memory – Operator interconnection network
Op
…
Op
Op
Operator – Operator interconnection network

Full crossbar — No constraint

ROMA network — Specific constraint

If $e_{ij_{ope\_ope}} = 1$ then $n_{j_{op}} = n_{i_{op}} + e_{ij_{op_r}}$

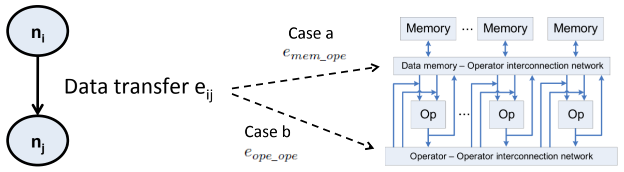
$e_{ij_{op_r}} :: \{1..\frac{|operators|}{2}\}$

Any point to point topology — Communication matrix

If $e_{ij_{ope\_ope}} = 1$ then $n_{j_{op}} = ComMat[n_{i_{op}}]$

Timing constraints

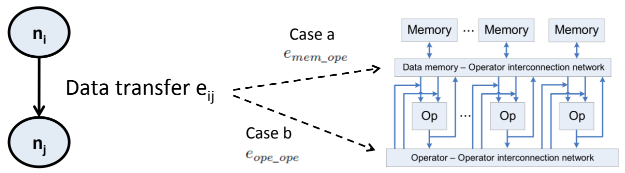
➢ Precedence constraints


n_i
n_j
Data transfer e_ij
Case a
e_mem_ope
Case b
e_ope_ope
Memory
…
Memory
Memory
Data memory – Operator interconnection network
Op
…
Op
Op
Operator – Operator interconnection network

$$n_{end} = n_{start} + n_{delay}$$

Timing constraints

➢ Precedence constraints


n_i
n_j
Data transfer e_ij
Case a
e_mem_ope
Case b
e_ope_ope
Memory
…
Memory
Memory
Data memory – Operator interconnection network
Op
…
Op
Op
Operator – Operator interconnection network

$$n_{end} = n_{start} + n_{delay}$$

Case a

$$n_{i_{start_{WR}}} \geq n_{i_{end}}$$
$$e_{ij_{start_{RD}}} \geq n_{i_{start_{WR}}} + \Delta e_{ij_{WR}}$$
$$\Delta e_{ij_{WR}} = WR_{lat} * e_{i,j_{mem\_ope}}$$
$$e_{ij_{start_{RD}}} + \Delta e_{ij_{RD}} = n_{j_{start}}$$
$$\Delta e_{ij_{RD}} = RD_{lat} * e_{i,j_{mem\_ope}}$$

Case b

$$\Delta e_{ij_{ope\_ope}} = n_{j_{start}} - n_{i_{end}}$$
$$\Delta e_{ij_{ope\_ope}} \geq ope\_ope_{lat} * e_{ij_{ope\_ope}}$$

Resource sharing constraints

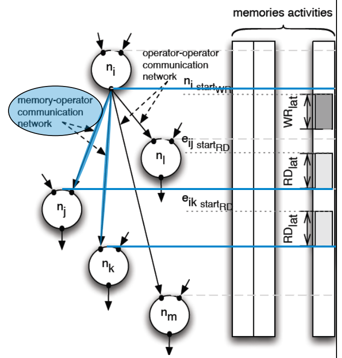
➢ Memory unit activity

Each operator has one data output port
=> 1 data is transfered

We model the potential memory operations as:

1 write operation / node
1 read operation / output edge
Same memory ID

<u>Exceptions</u>:
Input data => no write operation
Output data => no read operation



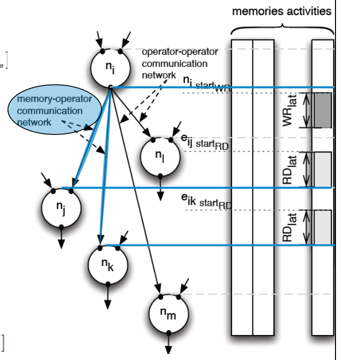
Resource sharing constraints

- Memory unit activity

$$Rec(n_{i_{WR}}) = [n_{i_{start_{WR}}}, n_{i_{mem}}, WR_{lat}, n_{i_{mem\_access}}]$$
$$n_{i_{mem\_acces}} \in \{0,1\} \Leftrightarrow \sum_{\forall e_{ij} \in n_{i_{outputs}}} e_{ij_{mem\_ope}} > 0$$
$$Rec(e_{ij_{RD}}) = [e_{ij_{start_{RD}}}, n_{i_{mem}}, RD_{lat}, e_{ij_{mem\_ope}}]$$

$$\texttt{Diff2}([...Rec(n_{i_{WR}}), Rec(e_{ij_{RD}})...])$$

$$\forall n_i \in E \wedge e_{ij}, e_{ik} \in n_{i_{outputs}} \wedge j \neq k$$
$$[Rec_j, Rec_k] = [Rec(e_{ij_{RD}}), Rec(e_{ik_{RD}})]$$



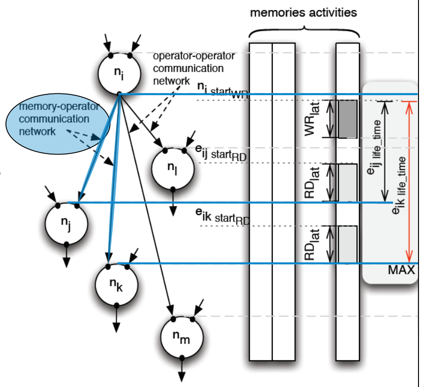
Resource sharing constraints

- Memory unit occupation

We consider that the data produced by node n_i transferred via a memory occupies a memory cell

from the start time of its write operation

until completion of the last read operation.

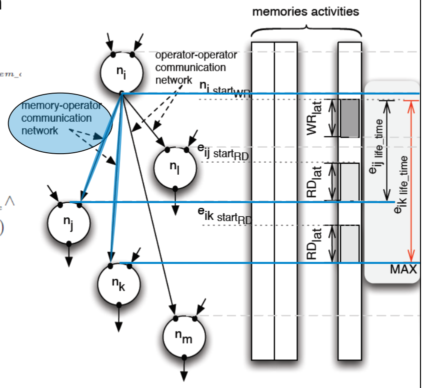


Resource sharing constraints

- Memory unit occupation

$$e_{ij_{life\_time}} = n_{j_{start}} - n_{i_{start_{WR}}}$$
$$n_{i_{life\_time}} = max(..., e_{ij_{life\_time}} * e_{ij_{mem\_}}$$
$$\texttt{where} \quad e_{ij} \in n_{i_{outputs}}$$

$$\forall m_i \in \{0..|Mem|-1\}, \forall n_i \in N :$$
$$(t_i = n_{i_{start_{WR}}} \wedge \Delta t_i = n_{i_{life\_time}} \wedge$$
$$m\_used_i :: \{0,1\} \Leftrightarrow n_{i_{mem}} = m_i)$$
$$\texttt{Cumulative}(t, \Delta t, m\_used, m\_size)$$



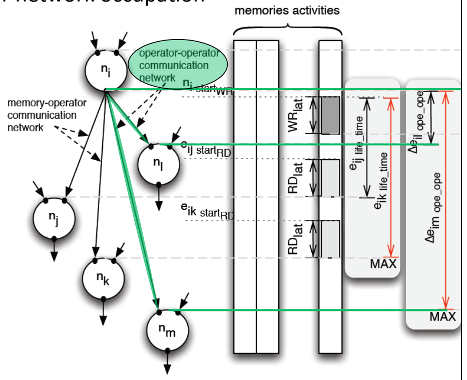
Resource sharing constraints

- Operator to operator network occupation

We consider that the operator to operator network is occupied for a data transfer

from the end of the execution of the node producing the data

until the last start of the sink nodes using this network

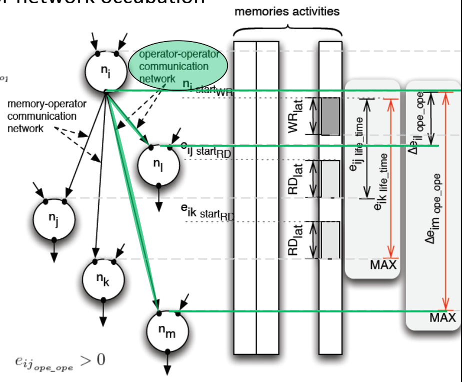


Resource sharing constraints

➢ Operator to operator network occupation

$$\Delta e_{ij_{ope\_ope}} = n_{j_{start}} - n_{i_{end}}$$

$$\Delta e_{ij_{ope\_ope}} \geq ope\_ope_{lat} * e_{ij_{ope\_o}}$$

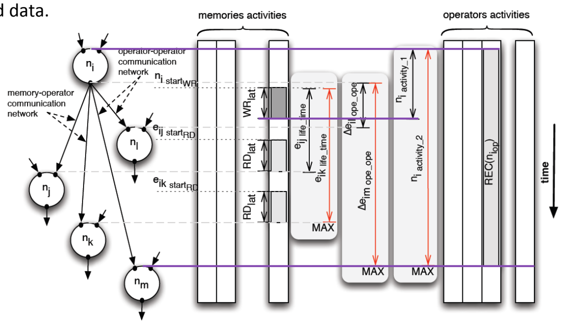


$$n_{i_{net\_acces}} \in \{0,1\} \Leftrightarrow \sum_{\forall e_{ij} \in n_{i_{outputs}}} e_{ij_{ope\_ope}} > 0$$

Resource sharing constraints

➢ Operator unit occupation

We consider an operator occupied <u>from</u> the start of the execution of the node representing the computed operation <u>until</u> the end of the last transfer of the produced data.



Resource sharing constraints

➢ Operator unit occupation

$$n_{i_{activity\_1}} = n_{i_{start_{WR}}} + WR_{lat} - n_{i_{start}}$$

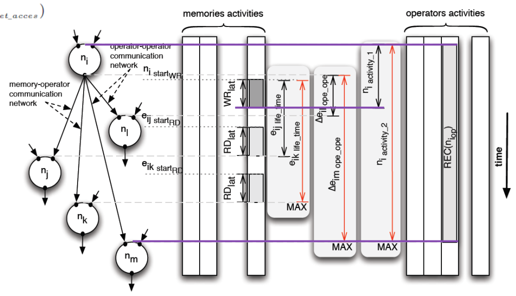
$$n_{i_{activity\_2}} = max(..., \Delta e_{ij_{ope\_ope}}, ...) + n_{i_{delay}}$$

where $e_{ij} \in n_{i_{outputs}}$

$$n_{i_{op_{activity}}} = max(n_{i_{activity\_1}} * n_{i_{mem\_acces}},$$
$$n_{i_{activity\_2}} * n_{i_{net\_acces}})$$

$$\forall n_i \in OPs : Rec(n_{i_{op}}) = [n_{i_{start}}, n_{i_{op}}, n_{i_{op_{activity}}}, 1]$$

$$\texttt{Diff2}([Rec(n_{1_{op}}), Rec(n_{2_{op}}), \ldots])$$



Cost function

➢ Minimization of the global computation time

$$CostFunc = max(..., n_{i_{end}}, ...)$$

Results – Applications from Mediabench

- In 78% of cases, our system provides results that are proved optimal

Application	DFG	nodes	edges	input nodes	output nodes	Cycles	Optimal	Runtime (ms)	Time Out (s)
JPEG IDCT (col)	1	35	40	13	4	16	yes	7693	30
-//-	2	57	65	22	5	26	yes	15117	30
Total DFGs for JPEG IDCT (col	1+2	92	105	35	9	42	yes	22810	30
JPEG IDCT (row)	3	106	127	34	17	29	no	TO	30
Write BMP Header	4	73	72	29	16	13	yes	875	10
-//-	5	19	18	8	4	5	yes	15	10
-//-	6	27	26	12	4	9	yes	47	10
-//-	7	27	26	12	4	9	yes	46	10
-//-	8	9	8	4	2	5	yes	0	10
Total DFGs for Write BMP Header	4+..+8	155	150	65	30	41	yes	983	10
sobel 7x7 (unrolled 2x2)	9	52	54	24	2	24	yes	360	10
MESA Matrix Mul	10	52	60	20	4	16	no	TO	30
IIR biquad N sections (unrolled x4)	11	66	73	29	1	55	no	TO	30
Roma H filter	12	43	42	21	2	28	yes	297	10

Application Graph Characteristics

Application	*AG*	nodes	edges	*IO* nodes
Auto Regression Filter	1	56	58	28
gost	2	21	21	11
Write BMP Header	3	71	70	43
-	4	19	18	12
-	5	27	26	16
-	6	27	26	16
-	7	9	8	6
total	3+..7	153	148	93
sobel 7x7	8	14	13	8
(unrolled 2x2)	9	49	51	24
MESA Matrix Mul	10	52	60	24
(unrolled x2)	11	88	120	32
(unrolled x3)	12	124	180	40
(unrolled x4)	13	160	240	48
IIR biquad N sections	14	18	19	9
(unrolled x4)	15	65	72	30
Roma H filter (unrolled)	16	42	41	22

Pipelined Execution Model

Latency and power consumption results for the pipelined architecture model with the ROMA operator patterns based library

AG	match (1 node match)	Latency	Nber of Op	Runtime (ms)	Optimal	Power (mW)	Nber of Op	Runtime (ms)	Optimal
1	28 (28)	24	28	374	yes	132	28	374	yes
2	10 (10)	6	10	78	yes	50	10	78	yes
3	28 (28)	11	28	561	yes	180	28	468	yes
4	7 (7)	2	7	16	yes	51	7	31	yes
5	11 (11)	5	11	15	yes	75	11	16	yes
6	12 (11)	5	11	15	yes	75	11	62	yes
7	3 (3)	2	3	0	yes	23	3	0	yes
3+..+7		25	60	607	yes	404	60	577	yes
8	7 (6)	15	5	16	yes	25	5	47	yes
9	25 (25)	27	25	280	yes	94	25	249	yes
10	28 (28)	12	28	375	yes	148	28	452	yes
11	56 (56)	12	56	1094	yes	296	56	1139	yes
12	84 (84)	12	84	2497	yes	444	84	2638	yes
13	112 (112)	12	112	5445	yes	592	112	5820	yes
14	9 (9)	18	9	62	yes	46	9	63	yes
15	36 (36)	60	36	515	yes	174	36	390	yes
16	20 (20)	34	20	390	yes	84	11	468	yes

Pipelined Execution Model (cont'd)

Latency and power consumption results for the pipelined architecture model with the UPaK patterns based library.

AG	match (1 node match)	Latency	Nber Of Op	Runtime (ms)	Optimal	Power (mW)	Nber Of Op	Runtime (ms)	Optimal
1	60 (28)	14	12	624	yes	x	x	TO	no
2	10 (10)	6	10	62	yes	52	10	63	yes
3	30 (28)	7	27	828	yes	180	26	546	yes
4	7 (7)	2	7	16	yes	51	7	16	yes
5	11 (11)	5	11	16	yes	75	11	16	yes
6	12 (11)	5	11	16	yes	75	11	32	yes
7	3 (3)	2	3	0	yes	23	3	0	yes
3+..+7		21	59	876	yes	404	58	610	yes
8	7 (6)	8	5	15	yes	33	5	31	yes
9	31 (25)	x	x	TO	no	123	19	608	yes
10	76 (28)	6	20	1219	yes	164	12	1156	no
11	152 (56)	6	40	2888	yes	332	24	2404	no
12	228 (84)	6	60	7007	yes	500	36	5884	no
13	304 (112)	6	80	17475	yes	668	48	13200	no
14	17 (9)	9	6	109	yes	50	6	187	yes
15	68 (36)	27	24	1686	yes	202	24	984	no
16	52 (20)	13	17	1223	yes	113	20	281	yes

Conclusion and futur work

➢ Conclusion

- We have presented a new **CP based system** to **solve simultaneously the scheduling, binding and routing** of a data flow graph on a **generic CGRA model**
- This system has been used to generate configurations for the ROMA processor
- Validation of this approach has been done by simulation at RTL level
- Design space exploration can be done using this system with a higher abstraction of the architecture model

➢ Future work

- Handle bigger problems thanks to a smart clusterization and a sliding windows
- …

Papers

- Raffin, E., Wolinski, Ch., Charot, F., Kuchcinski, K., Guyetant, S., Chevobbe, S., Casseau, E., *Scheduling, binding and routing system for a run-time reconfigurable operator based multimedia architecture*, In Proc. of Intl. Conf. on Design and Architectures for Signal and Image Processing (DASIP), Edinburgh, UK, Oct. 26-28, 2010 (Best Paper Award).

- Raffin, E., Wolinski, Ch., Charot, F., Kuchcinski, K., Casseau, E., Floch, A., Chevobbe, S., Guyetant, S. , *Scheduling, Binding and Routing System for a Run-Time Reconfigurable Operator Based Multimedia Architectur*e, IJERTCS International Journal of Embedded and Real-Time Communication Systems, vol. 3, no. 1, 2012, pp. 1-30.

Questions ?