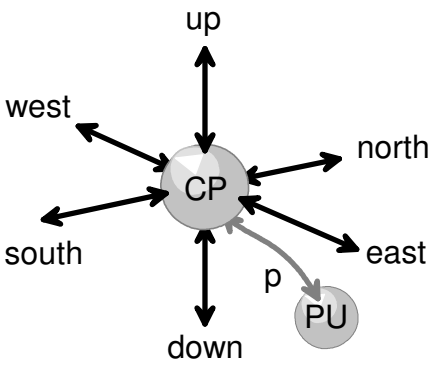
Gesellschaft zur Förderung angewandter Informatik e.V.

An- Institut an der Technischen Fachhochschule Berlin und der Fachhochschule für Technik und Wirtschaft Berlin
Mitglied der Arbeitsgemeinschaft industrieller Forschungsvereinigungen 'Otto von Guericke' e.V. (AiF)

Gerd Karl Heinz, Stephan Dengel, Carsten Busch, Vesselin Jossifov

-GFaI e.V. Berlin, FHTW Berlin-

"Fast, Adaptive Communication Processor for 2D/3D-Torus and Hypercube-Nets"



CP: Communication Processor
PU: Processing Unit

This project is supported by the BMWi (Bundesministerium für Wirtschaft) under registration 37/94; 1.11.93 - 31.3.95

Contact:

GFaI e.V.
Dr. G. Heinz
Haus 13.7
Rudower Chaussee 5
D-12484 Berlin

Tel. +49-30-6392 1600
Fax +49-30-6392 1602

Abstract

The main task of the project is to develop and to prove a circuit structure, defined by Krapp and Jossivof, that combines adaptive and deterministic packet routing for 2D-torus, 3D-torus and hypercube (HC) nets with the highest efficiency and with a minimum of gates.

Because the routing of 2D-, 3D-torus and hypercube nets uses a comparable algorithmic structure, it is possible, to create torus or hypercube nets with virtual dimensions and virtual links independent of eachother for each process.

Without software or micro-program usage, parallel automaton nets organize the routing and arbitration process. Thus, the communication is very fast. Typically, the router uses 4 clocks (in terms of the link data flow). After the destination address header, while a data packet of minimum 1 byte arrives the input FIFO, the routing process is complete.

Some Basic Ideas...

1] Jossifov, V.: Parallel Computer architecture's - theory and applications, Habilitation, Institut für Mathematik und Informatik der Bulg. Akad. der Wissenschaften, Dez. 1990, Sofia.

2] Krapp, M., Jossifov, V.: Formal specification of a distributed packet-routing algorithm with automation-nets, Technical report No.5, Bulgarian Academy of Sciences, Center for Informatics and Computer technology, Oct.1989, Sofia,

3] Krapp, M., Jossifov, V.: Formal specification of a distributed packet-router for **nD-hypercubes**. Workshop on parallel and distributed processing, Bulgarian Academy of Sciences, March,1990, Sofia, Elsevier Sci. Publ., 1991, accepted for presentation also on Joint Conference on Vector and Parallel Processing - CONPAR 90, Zurich, Sep, 1990 and at Workshop on Algorithms and Parallel VLSI Architectures, June 1990, Pont-a-Mousson.

4] Krapp, M, Jossifov, V.: Formal specification of a dynamic **load balancing** packet routing algorithm for binary **nD-hypercubes**, ZKI-Berlin Technical report 6/91.

5] Jossifov, V., Krapp, M.: Adaptive packet routing algorithm for **3D-meshes** described with parallel automaton nets, ZKI-Berlin Technical report 6/91.

6] Jossifov, V., Krapp, M.: **Adaptive routing** algorithm for **3D-mesh** computer networks - formal description with automaton nets. 36. Int. Wiss.Kolloquium der TU Ilmenau, 24.-28.10.1991, Ilmenau.

7] Jossifov, V.: The challenge of the key technologies in High-Performance Computing - Some aspects, Deutsch-Französisches Kolloquium an der FHTW Berlin "Moderne Rechnerarchitekturen", 07-09.12.1992, Berlin.

References

1] Tuazon,J., Peterson,K., Puiel,K., Lirbermann,D.: Caltech/IPC Marc II Hypercube concurrent processor, Proc of ICPP, 1985, pp. 666-673.

2] Jesshope,C.R.: Dynamic load balanced active data model of parallel processing for vision, Proc. BCS, Workshop on parallel processing for vision, Oxford University, 1986.

3] Jesshope,C.R.: Parallel processing based on active data, Proc. BIRA seminar on Parallel processing and super computing, Antwerpen, Nov.,1987.

4] Jesshope,C.R.:High performance Communication architectures, Proc. Workshop WP&DP'90, (Ed) K.Bojanov, North-Holland, march 1990, Sofia.

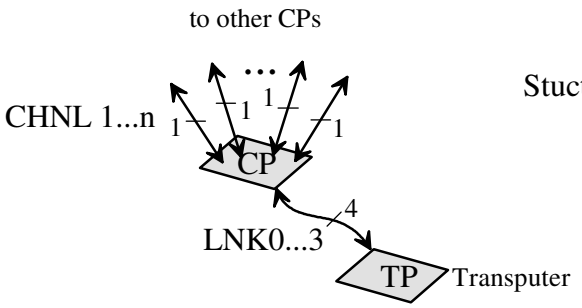
5] Jesshope,C.R.: Communications architectures for finegrain parallel processing, Proc. Workshop PARCELLA'90, Berlin, 1990.

6] Jesshope, C. R., Miller, P., Jantchev, J.:Programming with active data, Workshop PARCELLA'88, Berlin, 1988.

7] Getov,V., Jesshope,C.R.: Simulation facility of distributed memory system with "mad postmen" communication work, Proc. Sec. Europ. Distrib. Memory Comp. Conference, april 1991, Munich.

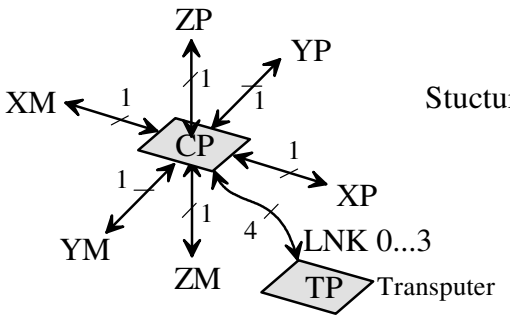
Net Structures

n-dim. Hypercube



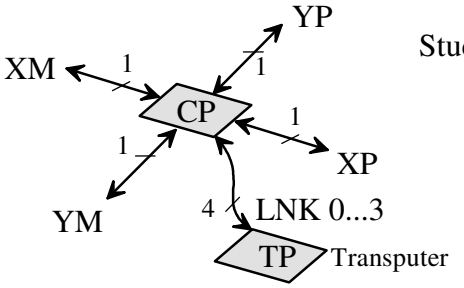
Stucture of the n-dim-HC

3-dim. Torus Net



Stucture of the 3d-Net Node

2-dim. Torus Net

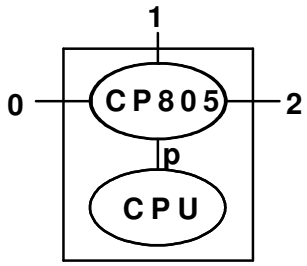


Stucture of the 2d-Net Node

Gesellschaft zur Förderung angewandter Informatik e.V. (GFaI)

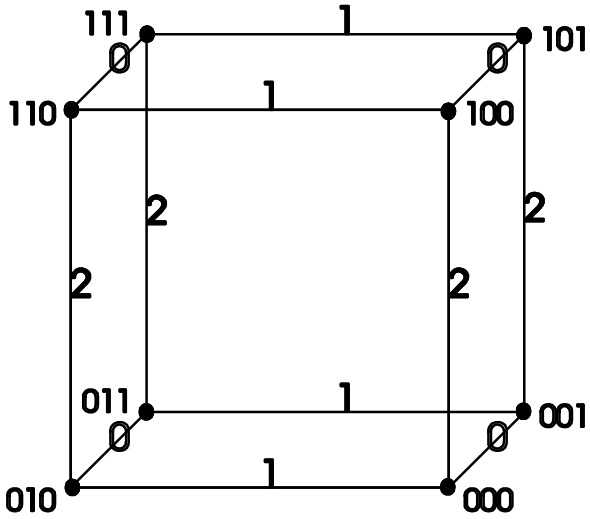
An- Institut an der Technischen Fachhochschule Berlin und der Fachhochschule für Technik und Wirtschaft Berlin
Mitglied der Arbeitsgemeinschaft industrieller Forschungsvereinigungen 'Otto von Guericke' e.V. (AiF)

Example *e-Cube* (Hyper Cube, HC)



Connections at a 3-dim. hypercube corner.
net-connected: 0, 1, 2
p: processor-connection.

(2,1,0): Changing bit positions



Node numbering scheme at a 3-dim. hypercube:
- Node numbers 000 bis 111 (bin.)
- directions := bit positions 0...2

Example a)

Routing Address Difference:

***rd* = {*dn*, *d*(*n*-1), ..., 2, 1, 0}; dn_i: binary for HC-nets**

Routing from node 7 to node 3:

rd = 111 ⊗ 011 = 100

The routing has to follow the edge 2 only.

Example b)

Routing from node 1 to node 6:

rd = 001 ⊗ 100 = 101

The routing has to follow the edges 2 and 0 or 0 and 2.

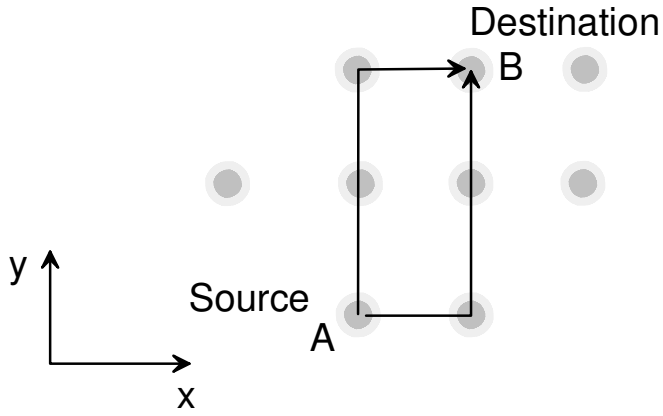
In terms: - adaptive routing: 2 after 0 or 0 after 2
- deterministic routing: one possibility only.

Gesellschaft zur Förderung angewandter Informatik e.V. (GFaI)

An- Institut an der Technischen Fachhochschule Berlin und der Fachhochschule für Technik und Wirtschaft Berlin
Mitglied der Arbeitsgemeinschaft industrieller Forschungsvereinigungen 'Otto von Guericke' e.V. (AiF)

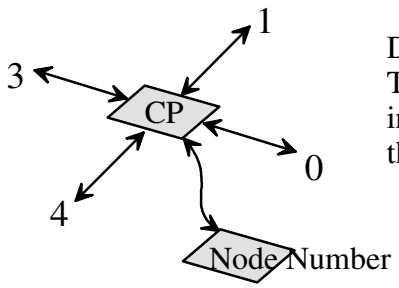
Routing 2D/2D toroidal nets

Example 2D-torus net:



Routing Address Difference: *rd*={*dx,dy*};

***dx,dy*: signed integer for 2D- and 3D-nets**



Directions at a 2D-node.
The CP has to compare only,
in which possible directions
the message can go.

Variation

Static routing

- with node enumberation: rd := const.
- logic comparison: $-1 \le dx \le +1$; $-1 \le dy \le +1$

Dynamic incr./decr. of the routing address

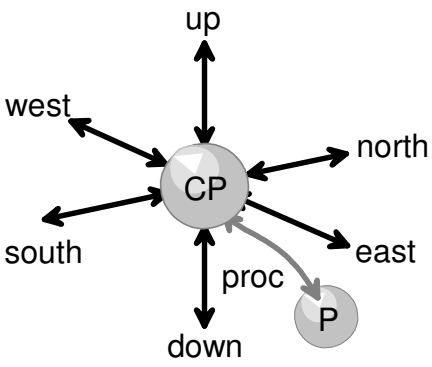
- without node enumberation: $rd := rf \pm 1$

- arithmetic decr./incr. of the destination address

Gesellschaft zur Förderung angewandter Informatik e.V. (GFaI)

An- Institut an der Technischen Fachhochschule Berlin und der Fachhochschule für Technik und Wirtschaft Berlin
Mitglied der Arbeitsgemeinschaft industrieller Forschungsvereinigungen 'Otto von Guericke' e.V. (AiF)

2D/3D- adaptives Routing



CP-Knoten im Detail.
Bezeichnung der sechs Richtungen am CP. Kanal p kennzeichnet vier Transputer-Links.

CP: Kommunikationsprozessor
P: Prozessor
proc: Prozessor- Links
up, down, north, west, east, south: Kanäle.

Zum 3d-Algorithmus gelten folgende Vereinbarungen:

- **Die Netzknoten sind fortlaufend mit einem Tupel *x,y,z* nummeriert.**
- **Jeder Kommunikationsprozessor kennt seine Knotennummer.**

Zum Verbindungsaufbau sendet der Quellprozessor den Zieltupel an den Kommunikationsprozessor in seinem Netzknoten. Dieser vergleicht den Zieltupel mit der ihm bekannten Netznummer. Daraus ist erkennbar, in welcher Richtung der Zielprozessor liegt und welche Ausgänge benutzt werden können.

Anschließend prüft der Kommunikationsprozessor, welcher Ausgang momentan frei ist und verzweigt die Verbindung zu diesen Ausgangskanal. Der Zieltupel wird vom Kommunikationsprozessor zum Folgeknoten gesand.

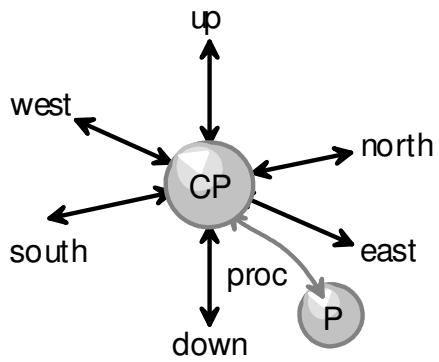
Ist der Zieltupel gleich der Knotentupel, ist der Zielknoten erreicht und der Kommunukationsprozessor verzweigt zum Zielprozessor.

Sollte ein Kommunikationsprozessor keinen freien Ausgang finden, wartet die Anforderung bis ein günstiger Ausgang frei wird.

Gesellschaft zur Förderung angewandter Informatik e.V. (GFaI)

An- Institut an der Technischen Fachhochschule Berlin und der Fachhochschule für Technik und Wirtschaft Berlin
Mitglied der Arbeitsgemeinschaft industrieller Forschungsvereinigungen 'Otto von Guericke' e.V. (AiF)

Arbiter 3D-Torus



1] priority of parallel occuring demands

2] which port is free?

3] set a locked channel busy

Priority-chain:

- 2D: K_2= {north, east, west, south, proc}, m=5
- 3D: K_3= {up, down,north, east, west, south, proc}, m=7

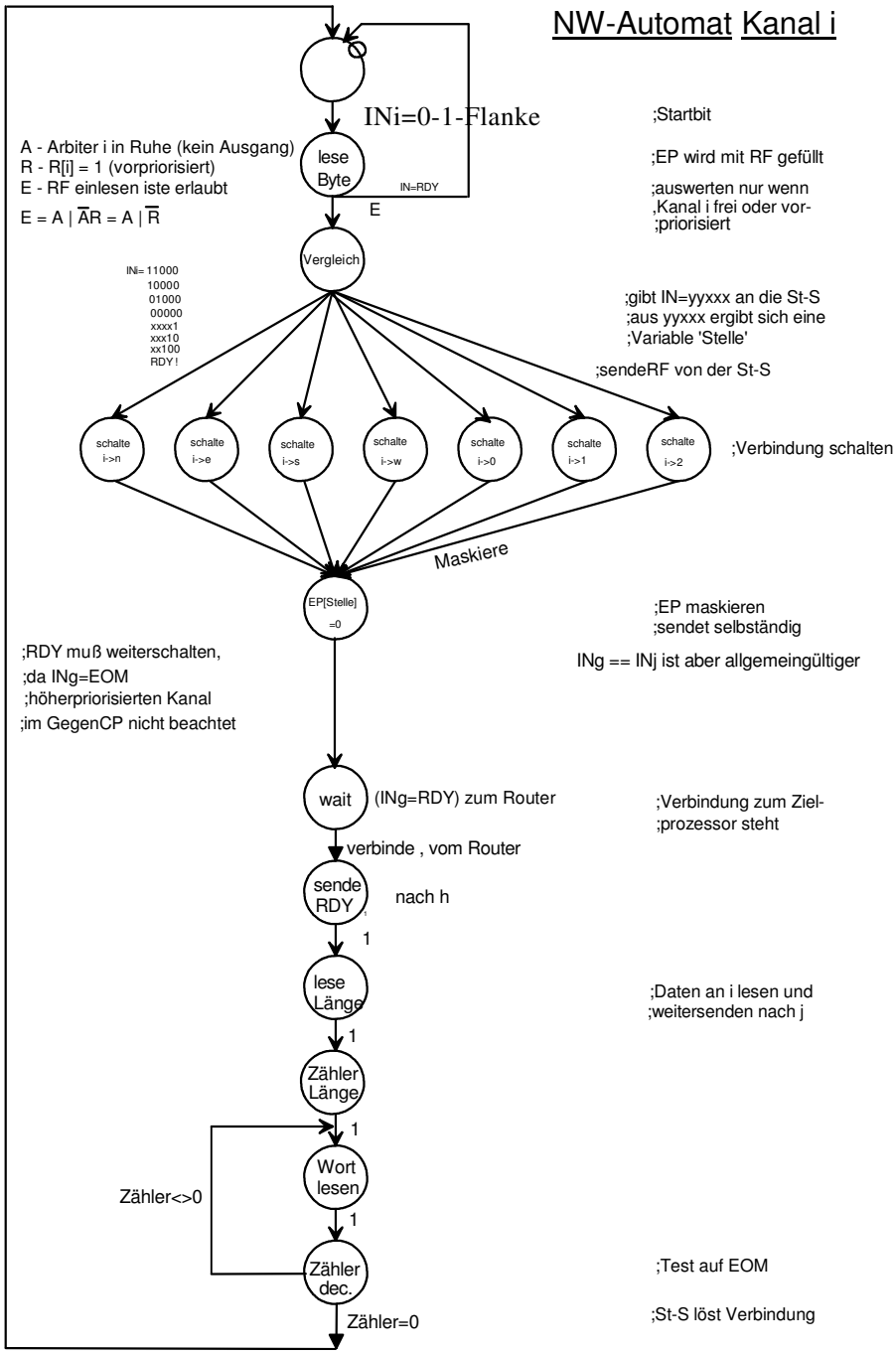
Equation:

$$free\ ij = req\ ij \bigwedge_{\substack{k=i+1 \\ k \neq j}}^{m} \overline{req\ kj} \bigwedge_{k=j+1}^{m} \overline{free\ jk}$$

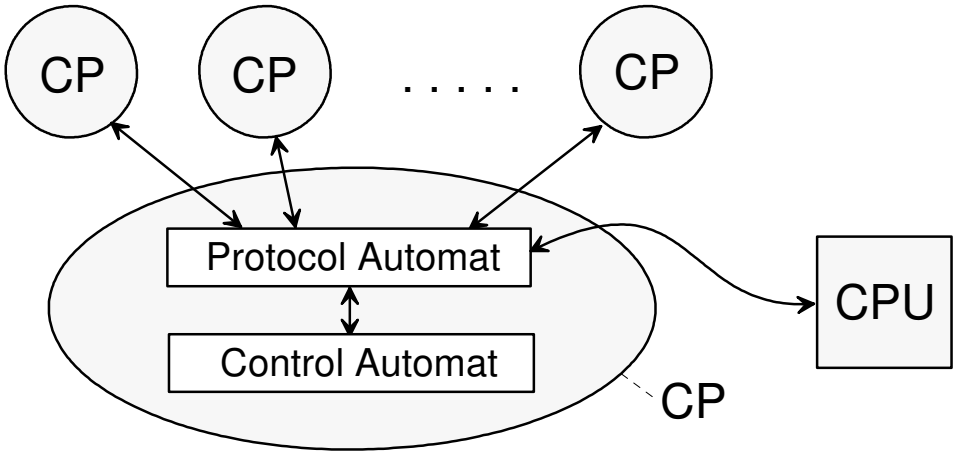
Gesellschaft zur Förderung angewandter Informatik e.V. (GFaI)

An- Institut an der Technischen Fachhochschule Berlin und der Fachhochschule für Technik und Wirtschaft Berlin
Mitglied der Arbeitsgemeinschaft industrieller Forschungsvereinigungen 'Otto von Guericke' e.V. (AiF)

Graph of the Protocol Automat



Basic Concept of the CP



Tasks:

Protocol Automat

- **Bit Synchronisation**
- **Word Synchronisation**
- **Frame Synchronisation**
- **Series to parallel and visa versa conversion**
- **Input- Buffer (FIFO)**

Control Automat

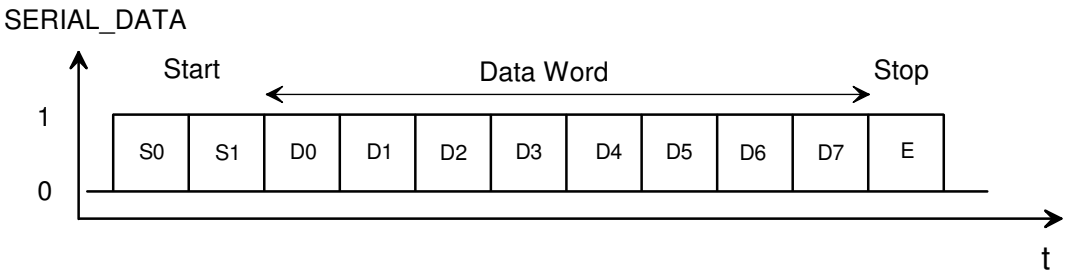
(Vergleicher, Maskierer, Router, Arbiter, Umsetzer)

- **Compare address and own node number**
- **Check the availability of possible output ports**
- **Priorize data flow to output ports**
- **Break waiting processes after time out (deterministic r.)**
- **Change output port, if busy (adaptive routing)**
- **Generate error messages**

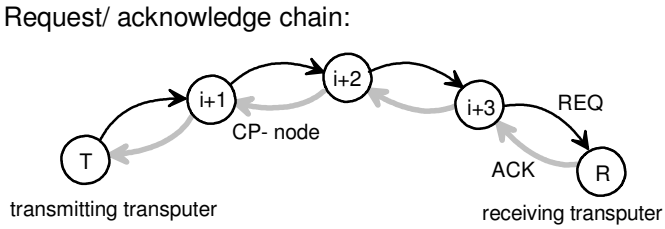
- **End of packet - detection**

Protocol

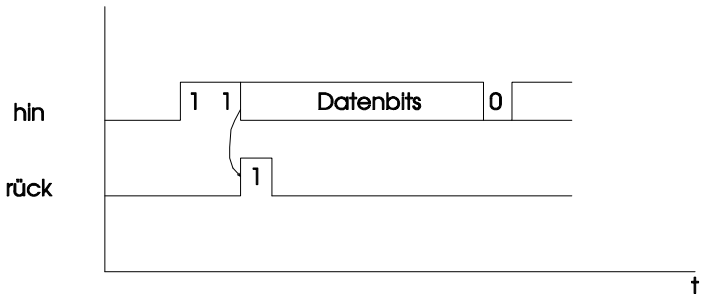
Word structure of Transputers INMOS T80x:



Acknowledging:



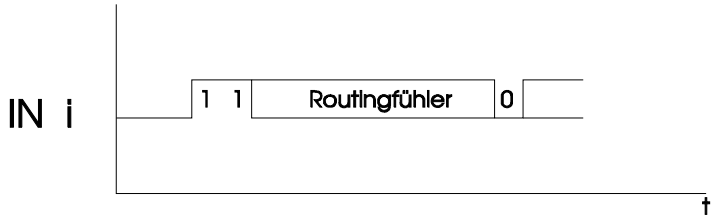
- request/acknowledge chain: usefull only to set wires busy
- multiple usage of wires not possible (only with C104-conv.)



Gesellschaft zur Förderung angewandter Informatik e.V. (GFaI)

An- Institut an der Technischen Fachhochschule Berlin und der Fachhochschule für Technik und Wirtschaft Berlin
Mitglied der Arbeitsgemeinschaft industrieller Forschungsvereinigungen 'Otto von Guericke' e.V. (AiF)

Routing address field:

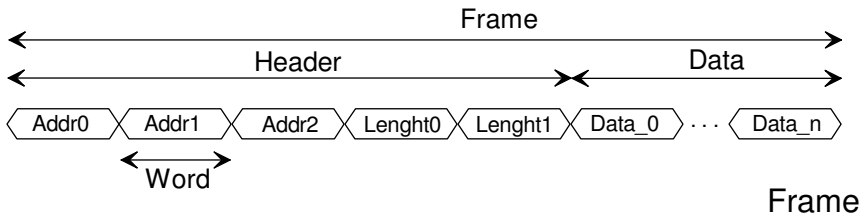


Gesellschaft zur Förderung angewandter Informatik e.V. (GFaI)

An- Institut an der Technischen Fachhochschule Berlin und der Fachhochschule für Technik und Wirtschaft Berlin
Mitglied der Arbeitsgemeinschaft industrieller Forschungsvereinigungen 'Otto von Guericke' e.V. (AiF)

Addressing 64k Nodes...

Frame Protocoll

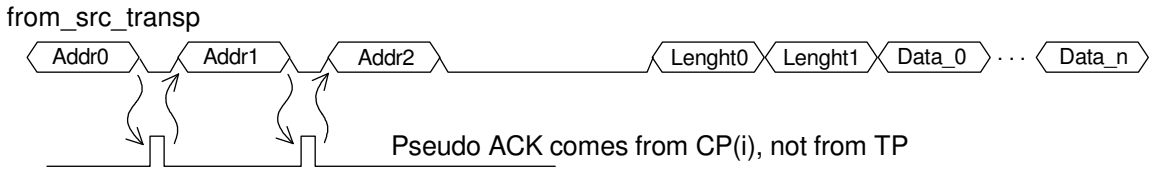


Frame Protocoll

Legende:

Addr0	: Ctl. word, Link No., Dest. node address0	Lenght0	: Number of following data words
Addr1	: Destination node address1	Lenght1	: Number of following data words
Addr2	: Destination node address2	Data_n ... Data_0	: Data words

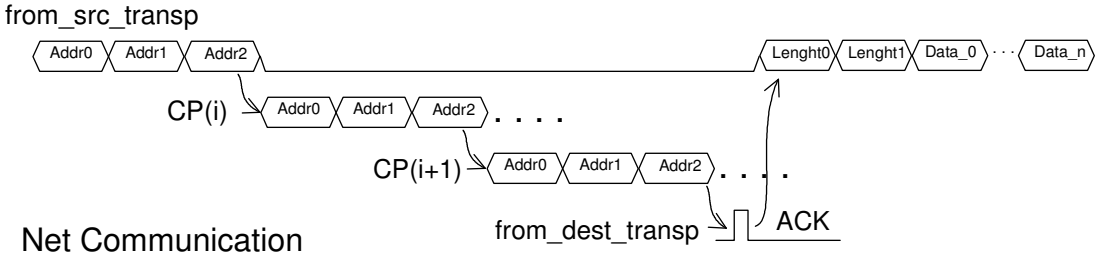
Pseudo Acknowledging:



Pseudo Acknowledging

To get the entire address from src_transp the connected CP generates two pseudo acknowledges.

Net Communication:



Net Communication

Gesellschaft zur Förderung angewandter Informatik e.V. (GFaI)

An- Institut an der Technischen Fachhochschule Berlin und der Fachhochschule für Technik und Wirtschaft Berlin
Mitglied der Arbeitsgemeinschaft industrieller Forschungsvereinigungen 'Otto von Guericke' e.V. (AiF)

Each CP has a delay typical of 50 gates. In dependence of the clock rate, one CP needs 2 to 10 clock transitions.

Gesellschaft zur Förderung angewandter Informatik e.V. (GFaI)

An- Institut an der Technischen Fachhochschule Berlin und der Fachhochschule für Technik und Wirtschaft Berlin
Mitglied der Arbeitsgemeinschaft industrieller Forschungsvereinigungen 'Otto von Guericke' e.V. (AiF)

Address Formats for Different Nettypes

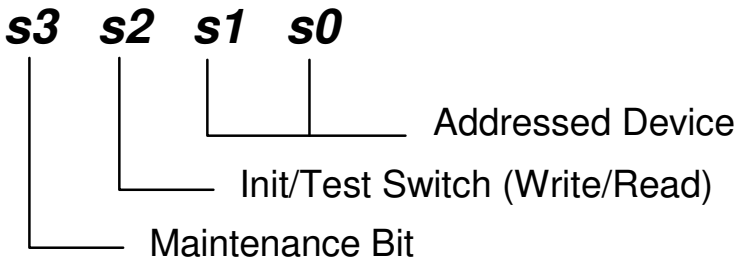
Address Format

Word	Addr0								Addr1								Addr2							
Function:	Control				Link		Addr		Addr								Addr							
Hypercube	s0	s1	s2	s3	n0	n1	a00	a01	a02	a03	a04	a05	a06	a07	a08	a09	a10	a11	a12	a13	a14	a15	a16	a17
3 dim. Net	s0	s1	s2	s3	n0	n1	x0	x1	x2	x3	x4	x5	y0	y1	y2	y3	y4	y5	z0	z1	z2	z3	z4	z5
2 dim. Net	s0	s1	s2	s3	n0	n1	x0	x1	x2	x3	x4	x5	x6	x7	x8	y0	y1	y2	y3	y4	y5	y6	y7	y8

Link Destination *n1,n0*

n1	n0	Link Connection to
0	0	LNK0
0	1	LNK1
1	0	LNK2
1	1	LNK3

Control Bits *s3,s2,s1,s0*



Gesellschaft zur Förderung angewandter Informatik e.V. (GFaI)

An- Institut an der Technischen Fachhochschule Berlin und der Fachhochschule für Technik und Wirtschaft Berlin
Mitglied der Arbeitsgemeinschaft industrieller Forschungsvereinigungen 'Otto von Guericke' e.V. (AiF)

Recognize the Port Address...

Kennung	Bedeutung
000	Ausgang über Kanal p
xx1	Ausgang über Kanal 0
x10	Ausgang über Kanal 1
100	Ausgang über Kanal 2
RDY	Bestätigung des Zielprozessors
EOM	des Quellprozessors

procedure ERKENNER

```
do
        switch ( IN(Kanal) )
                case 000 : Signal 000;
                case 100 : Signal 100;
                case x10 : Signal x10;
                case xx1 : Signal xx1;
                case EOM : Signal EOM;
                case RDY : Signal RDY;
while (1)
endprocedure ERKENNER
```

Equations for automata

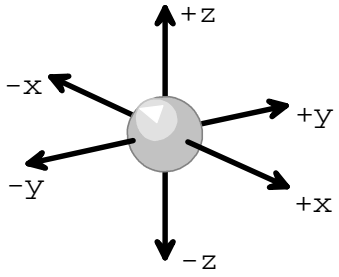
[xx1] = /sp ∩ D0
[x10] = /sp ∩ /D0 ∩ D1
[100] = /sp ∩ /D0 ∩ /D1 ∩ D2
[000] = /sp ∩ /D0 ∩ /D1 ∩ /D2
[RDY] = /sp ∩ /D0 ∩ /D1 ∩ /D2

Gesellschaft zur Förderung angewandter Informatik e.V. (GFaI)

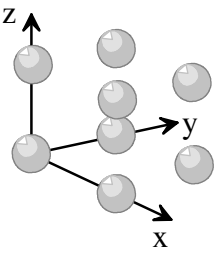
An- Institut an der Technischen Fachhochschule Berlin und der Fachhochschule für Technik und Wirtschaft Berlin
Mitglied der Arbeitsgemeinschaft industrieller Forschungsvereinigungen 'Otto von Guericke' e.V. (AiF)

Find the output port at 2D/3D Nodes...

(On condition, that nodes are enumberated)



Achsenrichtungen am 3D-Knoten

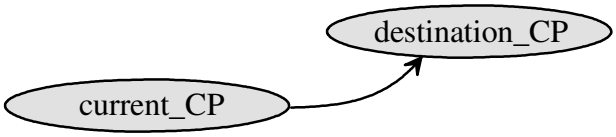


Bitbelegung:

```
x = [xn,...x1,x0]
y = [yn,...y1,y0]
z = [zn,...z1,z0]
```

Tupel der Zielknotennummer: T = [x,y,z]

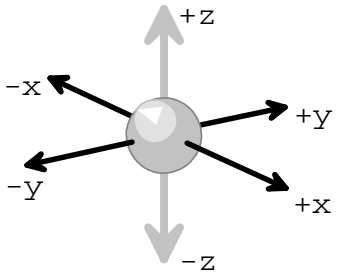
Beim 3D-Netz belegen die Vektoren
pro Achse mehrere Bitpositionen.



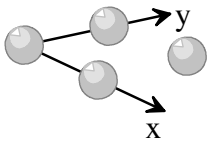
Zur Vermeidung von Addierern.
Für jede Achse x,y,z werden
gegenwärtige Knotennummer (Index c)
und Zielknotennummer (Index d)
in absteigender Reihenfolge verglichen.

Entscheidungsrichtung →

```
MSB .... LSB  MSB .... LSB  MSB .... LSB
xcn...xc0 ycn...yc0 zcn...zc0
xdn...xd0 ydn...yd0 zdn...zd0
```



Achsenrichtungen am 2D-Knoten



Bitbelegung:

```
x = [x0,x1,...xn]
y = [y0,y1,...yn]
```

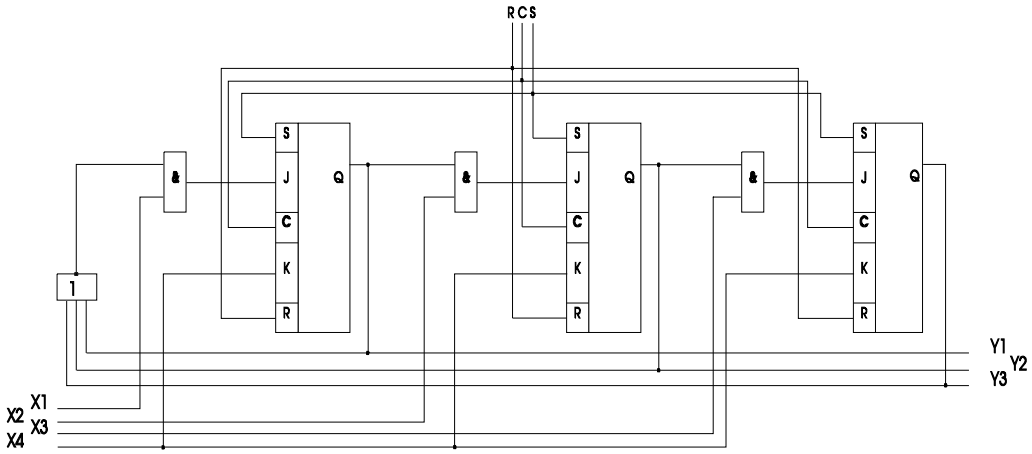
Beim 2D-Netz belegen die Vektoren
pro Achse mehrere Bitpositionen.

Gesellschaft zur Förderung angewandter Informatik e.V. (GFaI)

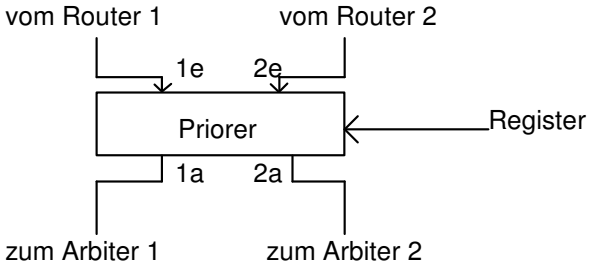
An- Institut an der Technischen Fachhochschule Berlin und der Fachhochschule für Technik und Wirtschaft Berlin
Mitglied der Arbeitsgemeinschaft industrieller Forschungsvereinigungen 'Otto von Guericke' e.V. (AiF)

Some Circuits

Router



Priorer



1a = 1e ∩ (reg[bit] ∪ /2e)
2a = 2e ∩ (/reg[bit] ∪ /1e)

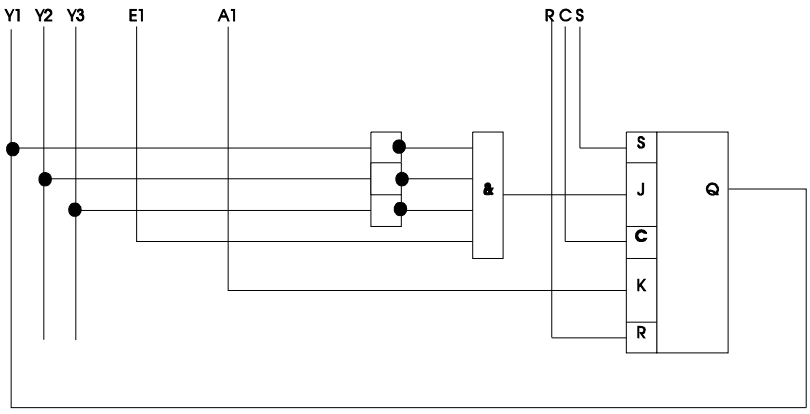
```
procedure PRIORER (Ausgang)
1e =  ∪ⁱ 'req_i' [Ausgang]          // ODER aller 'req'-Signale
2e =  ∪ⁱ 'free_j' [Ausgang]         // ODER aller 'free'-Signale
1a =   1e & (Register[Kanal] | !2e)   // C- Syntax: AND: &, OR: |, NOT: !
if (1a) return 1;                   //Ausgang frei
else return 0;                      //nicht frei
endprocedure PRIORER
```

Gesellschaft zur Förderung angewandter Informatik e.V. (GFaI)

An- Institut an der Technischen Fachhochschule Berlin und der Fachhochschule für Technik und Wirtschaft Berlin
Mitglied der Arbeitsgemeinschaft industrieller Forschungsvereinigungen 'Otto von Guericke' e.V. (AiF)

Arbiter 3D-HC

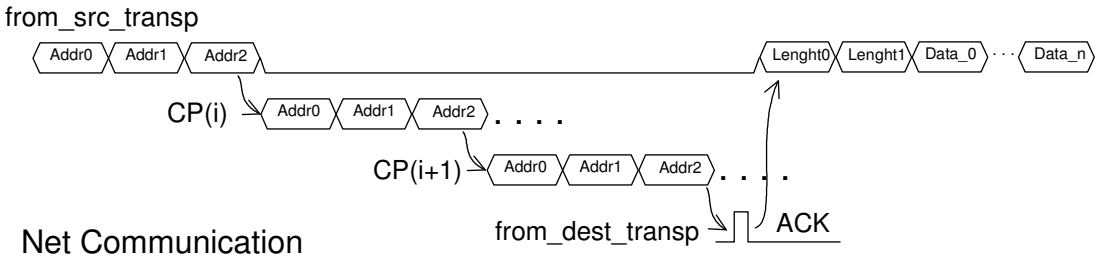
```
procedure ARBITER(Kanal,Ausgang)          // C- Programm Arbiter
while ( ! PRIORER(Ausgang));              //warten auf Priorer
free=0
while (! free) {
 switch (Kanal)
   case p:  switch (Ausgang)
                        case 0: free=req_0p & !(req_1p) & !(req_2p)
                        case 1: free=req_1p & !(req_2p)
                        case 2: free=req_2p
   case 0:  free=req_p0 & !(req_0p) & !(req_01)& !(req_02)
   case 1:  switch (Ausgang)
                        case p: free=req_p1 & !(req_01) & !(req_12) &
            !(req_1p)
                        case 0: free=req_01 & !(req_12) & !(req_1p)
   case 2:  switch (Ausgang)
                case p: free=req p2 & !(req_02) & !(req_12) & !(req_2p)
                case 0: free=req_02 & !(req_12) & !(req_2p)
                case 1: free=req_12 & !(req_2p)
 }
return free
endprocedure ARBITER
```



Gesellschaft zur Förderung angewandter Informatik e.V. (GFaI)

An- Institut an der Technischen Fachhochschule Berlin und der Fachhochschule für Technik und Wirtschaft Berlin
Mitglied der Arbeitsgemeinschaft industrieller Forschungsvereinigungen 'Otto von Guericke' e.V. (AiF)

Timing Results



Delay of the control circuits

Circuit	*Number of delays*	*Clocks*
router	2 flipflops, ...gates	2
arbiter	1 flipflop, ...gates	1

Delay of the protocol circuits (total)

Circuit	*Number of delays*	*Clocks*
bit-sync.	4 Flipflops, ...gates	4
inp.-buffer(3 + n) flipflops	3 + n	

Increasing total delay

***clk* no. of clocks**

***k* header address lenght**

***n* address vektor lenght**

clk = k + n + 10

Worst Case Example

18 bit address type: *k = 14*, *n = 18* (256k nodes addressable)

clk = 42

The algorithms scale linear with growing *n*. The lenght of the FIFO-buffer has to be greater then 42 bit (10 bit data). One CP delays the data flow typical for 10 clocks.

Summary

- support for 64k nodes in Single-Stage networks (back plane)
- minimal hardware expense (7000 gates for 3D-HC, 50.000 gates for 16D HC; C104: 1,2M Trs.)
- flexible and parallel routing of different net types
  - 2D/3D enumerated nets (address calculation integer)
  - 2D/3D non enumerated nets (address calculation binary)
  - hypercube (address calculation binary)
- variable expansion without changing the routing principles
  - deterministic
  - adaptive
- principle of node enumeration varies
  - absolute addresses (with node numbers)
  - relative addresses (dynamic difference)
- address calculation proved in the cases
  - dynamic address vector usage
  - static address vector usage
- simulations done at 2D_16, 3D_9, HC_8 have shown the function
- static coverage of wires (T80x) generates dead locks -> length limitation of messages is necessary
- arithmetical circuits avoidable (for HC and for 2D/3D too)
- total delay of one CP nearly 50 gates (quantization effects of clocks not considered)
- building blocks usable for both address types (integer, binary)
- restrictions came from the power consumption with increasing channels

- **differences in the number of ports (2D: 4; 3D: 6; nD-HC: n) are solvable with virtual link concepts only -> packet routing**
- **CP can handle virtual links, virtual processes and multitasking**