

**The Journal of
Mathematics and Computer Science**

Available online at

<http://www.TJMCS.com>

The Journal of Mathematics and Computer Science Vol. 4 No.2 (2012) 207 - 215

Novel Method for Blocking Diagnosis in Baseline Network

Peyman Forouhar*¹, Eiman Ghanbari²

¹ Islamic Azad University of Arak, Arak, Iran
peyman.foroohar@gmail.com

² Amirkabir University of Technology, Tehran, Iran
eiman.ghanbari@gmail.com

Received: January 2012, Revised: April 2012

Online Publication: June 2012

ABSTRACT

Multistage Interconnection Networks which are based on the connection switches are used to connect processors and memories in the parallel computer systems. These networks are classified into Blocking and Non Blocking Networks. In Blocking MINS, Blocking on the switches is very essential. Baseline network is one of the various Blocking MINS. If a permutation Factor is applied to a baseline network, and no problem is occurred in terms of blocking, we will regard that permutation as admissible otherwise it will be considered as inadmissible. In this paper, a method is presented for a quick and systematic diagnosis of an admissible permutation in the baseline network that can diagnose switches in case of applying an inadmissible permutation to a block network so that with the help of fault tolerance procedures, a way is prepared for complete and correct transmission of each desired permutation (PE). Hence, in this way, the most fundamental problem of the baseline network, that is its blocking will be diagnosed.

Keywords: Multistage Interconnection Networks, Baseline Network, Permutation Factor, Blocking.

INTRODUCTION

Today multistage interconnection networks utilized as one of the fastest connection channels and have variety operations in design of parallel computer systems. In fact, in

* Corresponding Author

this connection between source and destination nodes have been set from multi layers of switches [1]. How to connect the layers can be expressed different types of networks, multistage Interconnection networks to be divided into two general categories: static and dynamic networks. These networks are generally based on switches which are closed during the execution time and so, the path between source and destination nodes is created by these switches [2]. Dynamic networks based on the occurrence or non-occurrence of obstruction in the switches which is divided into two major types of blocking and non-blocking. baseline network is a multistage interconnection network , $BL(N*N)$ is baseline network with N inputs and outputs , this network has $\log N$ switch layers and each layer has $N/2$ switch [3]. a $2*2$ switch is a baseline network . This mechanism copies a baseline network with $N/2$ size under itself to build a larger baseline network and then adds an additional layer of switches on the input side Regularly, high-port switches are connected to upper bass line and lower port switches are connected to the lower layer of base line and the result of this method is a new $N*N$ baseline network [4]. block problem may be occur because Baseline network is a blocking network that there are different states which all entries cannot be simultaneously transferred to any desired outputs. In the other words, after establishing multiple connections, the other communication paths are blocked [5]. Figure 1, shows block problem which is introducing an $8*8$ baseline network.

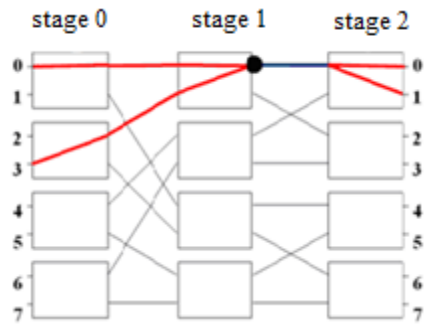


Figure1 : blocking problem in first switch of second layer in baseline($8*8$) network

BASIC DEFINITIONS

Defination1 (pattern): One pattern is defined for each layer in a $(N*N)$ BL network, so the total number of patterns are defined as a \log^N .

Definition 2 (lable): Label is used to determine the statuses of the switches which are closed.

Definition 3 (group): Group is used to delineate blocking or not blocking conditions for each switches.

Definition 4 (array): At each steps of running algorithm, method uses of several arrays. Thus $A(i)$ represents an array with the index i .

Definition 5 (switch): S_i is switches in each layer in network, index I is numeral of switch.

Definition 6 (pair input): Pair input is tow inputs which have been entered to the switch.

PSEUDO CODE IMPLEMENTATION of ALGORITHMS

Proposed algorithm for blockage detection are completed in \log^N steps in $(N*N)$ BL network. At the first step, the patterns are defined of obstruction and then status of all layers are specified in the other stages $(\log N - 1)$ by a loop

1- Determining patterns for each baseline network layer:

This stage uses the following function to define all patterns for each layer.

```

FUNCTION pattern (i) {
  NoG=2**i; SoG=N/NoG; temp=SoG; k=0;
  FOR(j=1;j<=NoG;j++){
    WHILE (k<SoG) {
      Group (j) ← k; k++;
    } SoG+= temp;
  } //End Of FOR
  Lable1 ← ALL group (o) IF o IS odd;
  Lable2 ← ALL group (e) IF e IS even;
} //END OF FUNCTION
    
```

Variables and symbols are used in this function:

NoG is numeral of used groups in this layer, SoG is size of group, I is number of layer, N is size of network, $A \leftarrow a$ is transmit a to set A.

2- Review the status of the first layer switches

At this stage, the switches status of the first layer will be studied in terms of occurrence or absence of obstruction occurrence and similarly, the closure of the switches has been reviewed by this function.

```

j=0;
FOR (i=0; i<N-1; i+=2)
  s(i) =  $\begin{pmatrix} a_i & a_{i+1} \\ b_i & b_{i+1} \end{pmatrix}$ ; C = 1;
  j++;
  FUNCTION set(s, pattern(c)) {
  IF s (i)  $_{o1} \in$  group (p) & s (i)  $_{o2} \in$  group (q)
    IF (p==q) RETURN(FALSE);
    ELSE IF s (i)  $_{o1} \in$  lable1
      SET s (i) to straight;
    ELSE IF s(i)  $_{o1} \in$  lable2
      SET s (i) to cross
  } // END OF FUNCTION
    
```

Variables are used in the above functions:

S(i) is switch with index I ,a and b are input and output the switch, c is number of pattern , $S(i)_{o1}$ is first output from switch with index I , and \in is mark of membership .

3- Determination of the second layer switches inputs and their status

According to the following code, this stage uses arrays and switches status of the first layer to survey second layer inputs.

```

i=0; p= 2**c; q=N/p;
FOR (j=0; j<q-1; j++) {
  IF (s (i) set IS straight) {
    
```

```

AI[J] =  $\begin{pmatrix} s(i)_{i1} \\ s(i)_{o1} \end{pmatrix}$  , A2[J] =  $\begin{pmatrix} s(i)_{i2} \\ s(i)_{o2} \end{pmatrix}$ 
}
ELSE IF (s (i) set IS cross) {
AI[J] =  $\begin{pmatrix} s(i)_{i2} \\ s(i)_{o2} \end{pmatrix}$  , A2[J] =  $\begin{pmatrix} s(i)_{i1} \\ s(i)_{o1} \end{pmatrix}$ 
}
}
i++;
} //END OF FOR

```

The variables are used in this above code:

A1, A2 is temp array, set is switch status, q is size of any temp array, p is number of needed temp array in this layer.

4- The next layers switch inputs and determine their status

This process release status of all layers and repeats recursive algorithm by using this layer inputs in this considered code.

```

j=0;
FOR (k=1; k<=p; k++)
  FOR (i=0; i<q-2; i+=2) {
    S (j) = (Ak [i] , Ak [i+1])
  }
j++;
}

```

A_k [i] is i'th cell from array A_k in last code .

EXAMPLE

The purpose of this paper is to find a method which it can be recognize allowable or non-allowable without drawing Network Map and routing. this algorithm end of after 4 stage .

First stage: Defining of patterns:

According to network size, the first stage defines patterns like figure2.

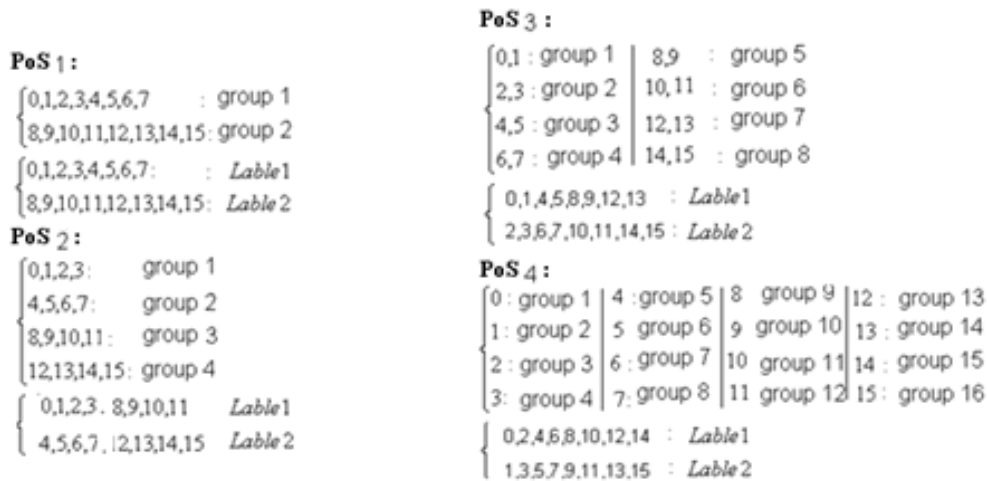


Figure 2: Defining of patterns for layers

Stage2: Review the status of the first layer switches:

At First, process converts permutations to the array in Figure 3 until all inputs of one switch come together such as 0, 1 that are first switch inputs and 2,3 are second switch inputs.

| | (S1) | (S2) | (S3) | (S4) | (S5) | (S6) | (S7) | (S8) | | | | | | | | |
|--------|------|------|------|------|------|------|------|------|----|---|----|----|----|----|----|----|
| arr-no | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| sw in | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| sw out | 6 | 12 | 1 | 9 | 11 | 3 | 4 | 14 | 10 | 2 | 5 | 13 | 7 | 8 | 15 | 0 |

Figure 3: inputs of first layer switches come together

In the first, two outputs of switch (which has been named to S (i)) should be reviewed. So necessary condition for this switch which isn't be blocked is that one of the first layer inputs exists in group (i) and the other exists in group (j) therefore this means $i \neq j$ in the first layer, for example outputs for S(1) are 6 and 12 , 6 member of grup1 and 12 member of grup2 then this switch (fist switch in first layer) is non blocking. By continuing this process and review all the switches can be concluded that the blockage does not occur for any of switches in the first layer. Furthermore, the method should be able to run apply this test to switches in the second layer so; it should determine Connection (closing) status for the first layer switches. This algorithm uses of labels to determine switches status this means that if this switch is member of lable1 this status is straight, but if it is a member of lable2, status is cross. . For example number 6 is a member of lable1 so switch status S(1) is straight. Repeat this algorithm for other S_i until survey all switches status in first layer so switches status show in figure 4 Closing status of the first layer switches in the network.

| | | | |
|-------|-------|-------|-------|
| S1: = | S2: = | S3: x | S4: = |
| S5: x | S6: = | S7: = | S8: x |

Figure 4 : switches status in first layer
'X' use for cross status and '=' use for straight status

Stage3: Determination of second layer switches inputs and their status

The purpose of this stage is putting the second layer switches inputs together like figure 3 for third layer. This work is based on the closing status of first layer switches like figure 4. At first make two array ,size of each of this array is half of array's size in figure 3. We fill

cells of tow array with similar index Simultaneously. For example, firstly we fill cells with index 0 of two arrays and secondly index 1 must be filled, This array must be based on the pair inputs of the first layer switches (Figure 3) and be completed by their closing status (Figure 4).Filling method uses of pair inputs (0, 1) of the first switch that is in straight mode (according to figure 4). insert 0 in index 0 of first array and insert 1 in index 1 of second and status for s2 switch is straight so pair input is (3,2) insert 2 in index 1 of first array and insert 3 in index 2 ,Inserting the first and second switches inputs in arrays is shown in (figure 5).

| | | arr.1 | | | | | | | | arr.2 | | | | | | | |
|--------|--|-------|---|---|---|---|---|---|---|-------|---|---|---|---|---|---|---|
| arr-no | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| sw in | | 0 | 2 | | | | | | | 1 | 3 | | | | | | |
| sw out | | 6 | 1 | | | | | | | 12 | 9 | | | | | | |

Figure 5: Inserting the first and second switches inputs in arrays

Switch status S_4 (figure4) is Cross and pair input is (5, 4) in this status, inserting the components to arrays is different with straight status this is shown in figure 6.

| | | arr.1 | | | | | | | | arr.2 | | | | | | | |
|--------|--|-------|---|---|---|---|---|---|---|-------|---|----|---|---|---|---|---|
| arr-no | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| sw in | | 0 | 2 | 5 | | | | | | 1 | 3 | 4 | | | | | |
| sw out | | 6 | 1 | 3 | | | | | | 12 | 9 | 11 | | | | | |

Figure 6: Inserting the tertiary switches inputs in arrays

This will continue until all places are filled by numbers in each array (figure 7).

| | | arr.1 | | | | | | | | arr.2 | | | | | | | |
|--------|--|-------|------|------|------|------|------|------|------|-------|---|----|----|----|----|----|----|
| | | (S1) | (S2) | (S3) | (S4) | (S5) | (S6) | (S7) | (S8) | | | | | | | | |
| arr-no | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| sw in | | 0 | 2 | 5 | 6 | 9 | 10 | 12 | 15 | 1 | 3 | 4 | 7 | 8 | 11 | 13 | 14 |
| sw out | | 6 | 1 | 3 | 4 | 2 | 5 | 7 | 0 | 12 | 9 | 11 | 14 | 10 | 13 | 8 | 15 |

Figure 7: inputs of second layer switches come together

All outputs are corresponding with all of inputs in array1 and array2. Obviously, the form is obtained in Figure 7, is similar to the form in Figure 3. This algorithm diagnoses Occurrence or non-occurrence of obstruction for any switches status by repeating second and third stages furthermore with the defined patterns for third layer. At this stage, for non-blocking of each second layer switches, outputs shouldn't be member s of one of four PoS₂ groups simultaneously and one of outputs member of group(i) and other member of group(j) and i≠j. Operation is according to previous stage if blocking problem for two states of switch (cross and straight) doesn't occur. At end of this phase, blocking problem in second layer don't occur which is displayed in the figure 8.

| | | | |
|-------|-------|-------|-------|
| S1: x | S2: = | S3: = | S4: x |
| S5: x | S6: = | S7: = | S8: = |

Figure 8: switches status in second layer

Stage4: survey inputs and status next layer switches:

Each other layers like third layer uses of pattern and repeats all stages for surveying switches. This is important that each array is divided into two arrays and uses of these arrays with static pattern for surveying switch in each layer. The result of for tertiary layer is shown in in figure s 9, 10 and for fourth layer is shown in figure 11 and figure12.

| | arr.1 | | | | | | | | arr.2 | | | | | | | |
|--------|---------|------|------|------|---------|------|------|------|---------|----|----|----|---------|----|----|----|
| | arr.1-1 | | | | arr.1-2 | | | | arr.2-1 | | | | arr.2-2 | | | |
| | (S1) | (S2) | (S3) | (S4) | (S5) | (S6) | (S7) | (S8) | | | | | | | | |
| arr-no | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 |
| SW in | 2 | 5 | 9 | 15 | 0 | 6 | 10 | 12 | 3 | 4 | 8 | 13 | 1 | 7 | 11 | 14 |
| SW out | 1 | 3 | 2 | 0 | 6 | 4 | 5 | 7 | 9 | 11 | 10 | 8 | 12 | 14 | 13 | 15 |

Figure 9: inputs of tertiary layer switches come together

| | | | |
|-------|-------|-------|-------|
| S1: = | S2: x | S3: x | S4: = |
| S5: = | S6: x | S7: = | S8: = |

Figure 10 : switches status in tertiary layer

| | arr.1 | | | | arr.2 | | | |
|--------|---------|-------|---------|-------|---------|-------|---------|-------|
| | arr.1-1 | | arr.1-2 | | arr.2-1 | | arr.2-2 | |
| | 1-1-1 | 1-1-2 | 1-2-1 | 1-2-2 | 2-1-1 | 2-1-2 | 2-2-1 | 2-2-2 |
| | (S1) | (S2) | (S3) | (S4) | (S5) | (S6) | (S7) | (S8) |
| arr no | 0 1 | 0 1 | 0 1 | 0 1 | 0 1 | 0 1 | 0 1 | 0 1 |
| sw in | 2 15 | 5 9 | 6 10 | 0 12 | 3 13 | 4 8 | 1 11 | 7 14 |
| sw out | 1 0 | 3 2 | 4 5 | 6 7 | 9 8 | 11 10 | 12 13 | 14 15 |

Figure 11: inputs of fourth layer switches come together

| | | | |
|-------|-------|-------|-------|
| s1: x | s2: x | s3: = | s4: = |
| s5: x | s6: x | s7: = | s8: = |

Figure 12 : switches status in fourth layer

It is noteworthy that blocking or non-blocking problem is occurred before Distinguishing of switches status. Switches status Can be diagnosed only in situation where is not blocked in previous layer.

CONCLUSION

In this paper, a method is presented for a quick and systematic diagnosis of an admissible permutation in the baseline network that can diagnose switches in case of applying an inadmissible permutation to a block network so that with the help of fault tolerance procedures with this method detect blocking switch this use for complete and correct transmission of each desired permutation (PE).this method shown with 16 channel network.

REFERENCES

- [1] R.Ferreira, M.Laure, A.C. Beck, T.Lo, M.Rutzig and L.Carro, A Low Cost and Adaptable Routing Network For Reconfigurable Systems, *Proc. of the IEEE International Symposium on Parallel & Distributed Processing*, pp. 1-8, 2009.
- [2] T.P.Troudet and S.M.Walters, Neural Network Architecture for Crossbar Switch Control , *IEEE Journals*, Vol. 38, pp. 42 - 56, 1991.
- [3] G.Hou and Y.Yang, SuperRecursive Baselines: A Family of New Interconnection Networks with High Performance/Cost Ratios Parallel Architectures, *Proc. of the IEEE*, Vol.10,pp.260-265, 2000.

- [4] Y.Yang and J.Wang, Routing Permutations on Baseline Networks with Node-disjoint Paths, *Parallel and Distributed Systems* , Vol.16, pp.737-746, 2005.
- [5] C.Lombardi, W.Feng and K. Huang, Detection and Location of Multiple Faults in Baseline Interconnection Networks, *IEEE Transactions on Computers*, Vol. 41, pp. 1340-1344, 1992.