Integer Linear Programming formulation for Table Placement

NOTE: This document is work in progress

Table of Contents

[Overview 3](#_Toc508176790)

[Full Crossbar 3](#_Toc508176791)

[ILP Formulation for full crossbar 3](#_Toc508176792)

[Segment Crossbar 6](#_Toc508176793)

[ILP Formulation for Segment crossbar 6](#_Toc508176794)

## Overview

Objective of Table placement is to place all tables in available SOMs honouring all constraints. There are constraints both from the schedule and availability of hardware resources. This problem can be solved using Integer Linear Programming(ILP). We have three types of crossbars available

1. Unit Crossbar
2. Full Crossbar
3. Segment Crossbar

We formulated ILP constraints for Full Crossbar and Segment Crossbar. For more details about Crossbars, refer DRMT paper. As a prerequisite, high level understanding of SOM and its structure is required to understand the constraints.

## Full Crossbar

### ILP Formulation for full crossbar

#### Input Data

Below are the inputs for defining constraints

|  |  |  |
| --- | --- | --- |
| No | Input Data | Definition/Description |
| 1 | S | number of SOM clusters |
| 2 | T | number of tables |
| 3 | NSR[i] | number of SRAMs needed by ith table |
| 4 | NTC[i] | number of TCAMs needed by ith table |
| 5 | NSRinSOM[j] | number of SRAMs available in jth SOM cluster |
| 6 | NTCinSOM[j] | number of TCAMs available in jth SOM cluster |
| 7 | NKSEG[i] | number of K segments needed by ith table |
| 8 | NDSEG[i] | number of D segments needed by ith table |
| 9 | NKSEGinSOM[j] | number of K segments available in jth SOM cluster |
| 10 | NDSEGinSOM[j] | number of D segments available in jth SOM cluster |
| 11 | NCT[i][c] | number of controllers of type *c* which are needed by ith table. |
| 12 | NCTinSOM[j][c] | number of controllers of type *c* available in jth SOM cluster. ‘*c*’ would be ReadController, WriteController, HashController etc. |
| 13 | {disjointTablesSet} | we have disjoint sets of tables. Each disjoint set signifies that those are the tables which would be accessed in a particular clock cycle by all drmt processors. |
| 14 | NUMSEGDBUS | number of segments in data bus. This bus connects all data outputs of each SOM cluster |
| 15 | NTCAMWIDTH[j] | number of tcams columns in TCAM grid of jth SOM cluster |
| 16 | NTCAMDEPTH[j] | number of tcams rows in TCAM grid of jth SOM cluster |

#### Variables

Following are variables defined in ILP

1. TS[i][j]: indicator variable to specify if ithtable can be placed in jth SOM cluster.
   1. Type: Boolean
   2. Range: 0 or 1
2. N[j]: variable to store number of tables stored in jth SOM cluster.
   1. Type: integer
   2. Range: 0 <= N[j] <= T

#### Constraints

Following are the constraints

1. Number of tables in an SOM cluster is the sum of all tables placed in that SOM. This defines constraint between TS and N ilp variables
2. Every table should be placed in one and only one SOM cluster
3. Sum of SRAMs used by tables placed in that SOM cluster should not exceed available SRAMs in that SOM
4. Sum of TCAMs used by tables placed in that SOM cluster should not exceed available TCAMs in that SOM
5. For each SOM, sum of number of Kseg used by tables in each disjointTableSet (if placed in that SOM cluster) should be less than or equal to number of Kseg available in that SOM cluster.
6. For each SOM cluster, sum of number of Dseg used by tables in each disjointTableSet (if placed in that SOM cluster) should be less than or equal to number of Dseg available in that SOM cluster.
7. For each controller type, for each SOM cluster, sum of number of that controllers of that type used by tables in each disjointTableSet (if placed in that SOM cluster) should be less than or equal to number of controllers of that type available in that SOM cluster.
8. Number of tables (which uses TCAM) that can be placed in a SOM is less than or equal to number of available TCAM Controllers

##### TCAM placement constraints

There are few constraints for placing the tables (which uses TCAMs) physically on to the TCAM grid.

Based on table key width, depth (number of entries), it is divided in to blocks (each of TCAM size)  
for eg, table of 3 x 2 is divided as below. Each block (A, B, C etc.,) will fit into a TCAM unit  
+------+------+------+  
| A | B | C |  
+------+------+------+  
| D | E | F |  
+------+------+------+

Blocks in each row are bitwise blocks. Column wise blocks are called word wise blocks.  
Each of the block A, B, C etc., is placed at (x, y) location of TCAM grid  
Below are constraints, A(x) gives x coordinate in TCAM grid, A(y) gives y coordinate in TCAM grid it is placed in  
A(x) = B(x) = C(x) -> all bitwise blocks should be allocated to same row of TCAM (i.e., x coordinate)  
D(x) = E(x) = C(x) -> all bitwise blocks should be allocated to same row of TCAM (i.e., x coordinate)  
A(y) < B(y) < C(y) -> All word-wise blocks of same bitwise row should be allocated different columns of TCAM  
D(y) < E(y) < F(y) -> All word-wise blocks of same bitwise row should be allocated different columns of TCAM  
if (A(x) = D(x)) then C(y) < D(y) -> if two word-wise blocks are allocated in same row of TCAM, then columns should not be same

Let us try to define some variables and constraints for the same.

#### Variables

1. R[i][j][k][l]: variable stores x coordinate of *‘(k, l)*’ block for table ‘*i*’ in SOM cluster ‘*j*’. From the above example A -> (0,0) block, B-> (0,1) block, C-> (0,2) block, D -> (1,0) block etc. Also say we have divided table into K \* L blocks.
   1. Type: Integer
   2. Range: 0 < R[i][j][k][l] < NTCAMDEPTH[j]
2. C[i][j][k][l]: variable stores y coordinate of *‘(k, l)*’ block or table ‘*i*’ in SOM cluster ‘*j*’. From the above example A -> (0,0) block, B-> (0,1) block, C-> (0,2) block, D -> (1,0) block etc. Also say we have divided table into K \* L blocks.
   1. Type: Integer
   2. Range: 0 < C[i][j][k][l] < NTCAMWIDTH[j]

#### Constraints

1. All bitwise blocks of a table should be placed in the same row of TCAM grid provided that table uses TCAM
2. Blocks in each bitwise should be placed in different column of TCAM grid provided that table uses TCAM
3. If two word-wise blocks are placed in same row of TCAM grid, then they should occupy different column in TCAM grid
4. If two tables placed in same SOM cluster, then blocks of those tables should not be placed in same TCAM of TCAM grid

Once we solve for these constraints, we have the needed solution in the indicator variable TS[i][j] i.e., which table is placed in which SOM cluster.

## Segment Crossbar

### ILP Formulation for Segment crossbar

We need the same inputs as needed by full crossbar ilp formulation. In **addition** to variables/constraints defined in ilp formulation for full crossbar, we need below.

Following are additional variables defined in ILP

1. SKT[j][k][i]: indicator variable to specify if ith table is assigned to kth segment of jth SOM cluster
   1. Type: Boolean
   2. Range 0 or 1

Following are the additional constraints

1. On a particular SOM cluster and particular segment, not more than one table should be assigned
2. For each SOM cluster, sum of number of segments to which a particular table is assigned is equal to number of K segments needed for that table