Combinatorial Optimization in Mapping Generalized Template Matching onto Reconfigurable Computers

Xuejun Liang

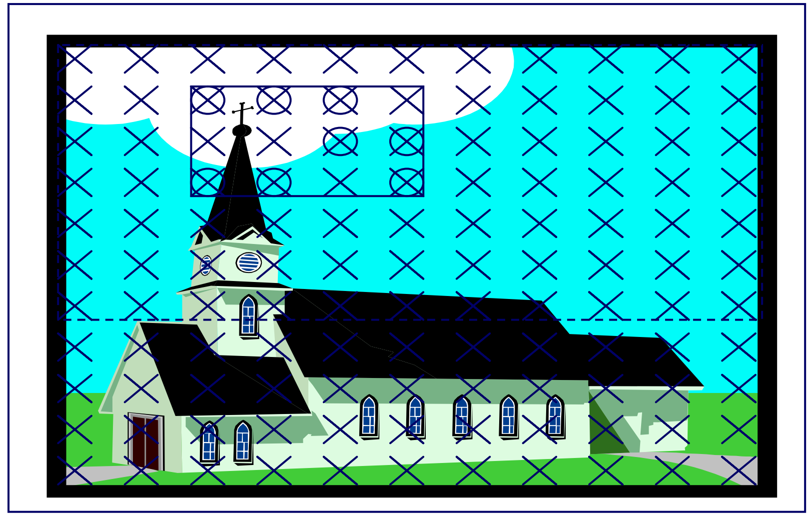
Department of Computer Science, Jackson State University, Jackson, MS, USA 39217

Qutaibah Malluhi

Computer Science and Engineering Department, Qatar University, Doha, Qatar

Generalized Template Matching (GTM)

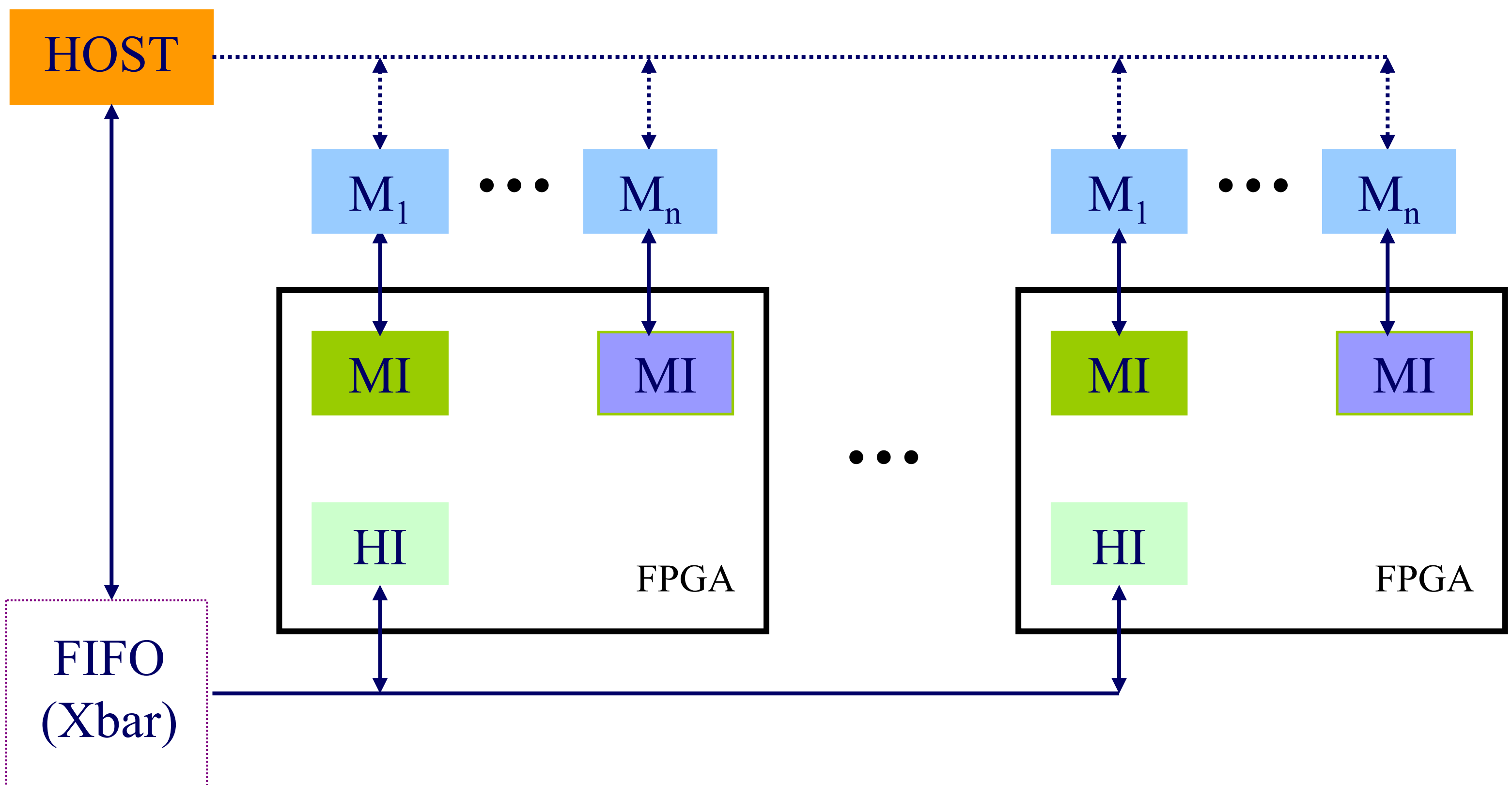
- Template Matching
- Two-D Digital Filtering
- Morphologic Operation
- Motion Estimation



For image region $i \leftarrow 1$ to r
    For template $j \leftarrow 1$ to $m(i)$
        For (all) pixel P in IR_i
            Basic-Function($P, T_{i,j}$)

Pixel level parallelism
Template level parallelism

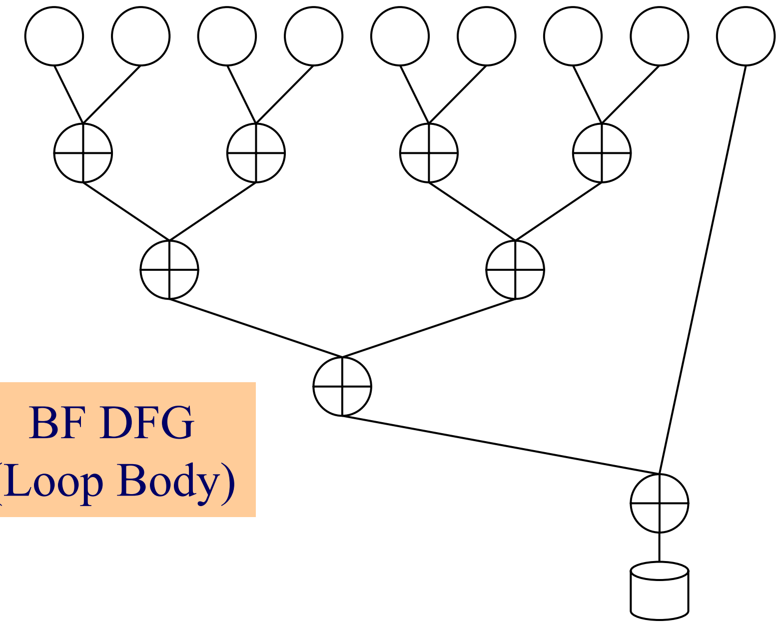
Target FPGA Board Architecture



GTM Mapping Procedure

GTM Application Specification

Image Regions	Templates
IR_1	$\{T_{1,1}, T_{1,2}\}$
IR_2	$\{T_{2,1}, T_{2,2}, T_{2,3}\}$
IR_3	$\{T_{3,1}, T_{3,2}\}$

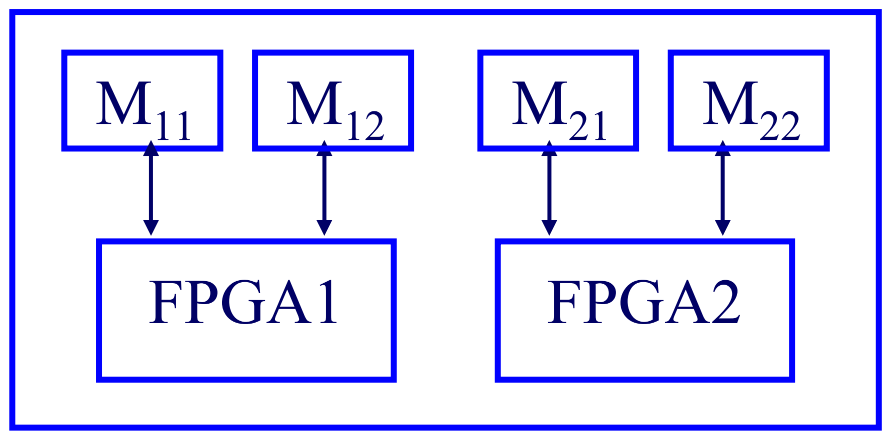


BF DFG
(Loop Body)

Component Library

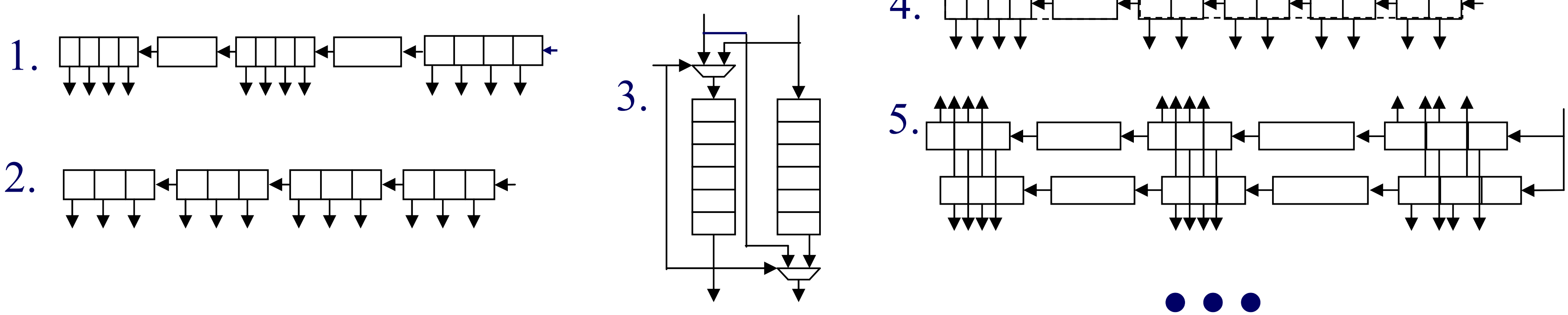
Adder, Multiplier,…

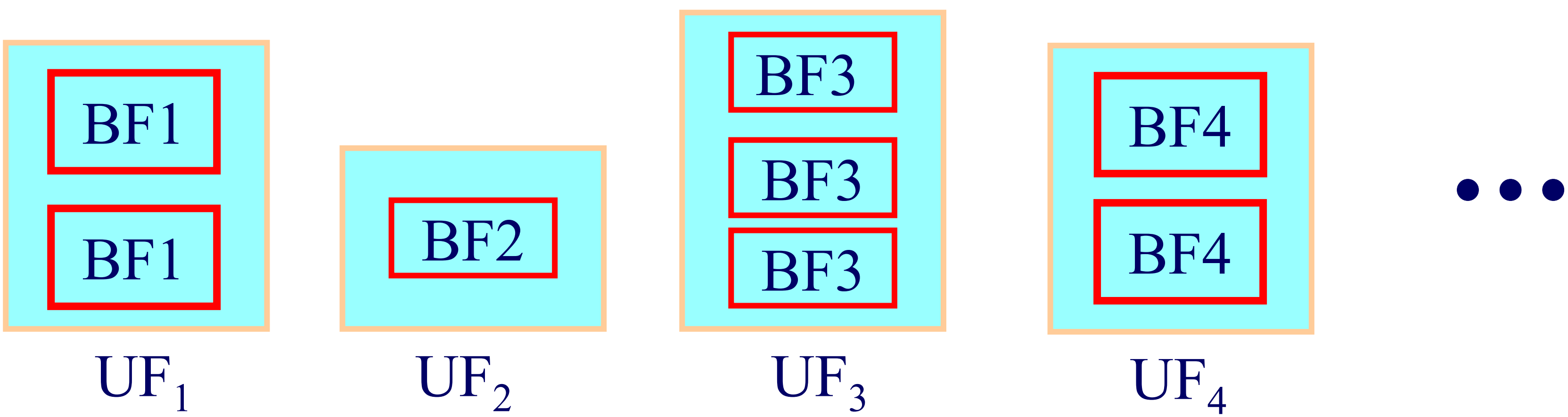
FPGA Board



GTM Mapping Procedure (Cont.)

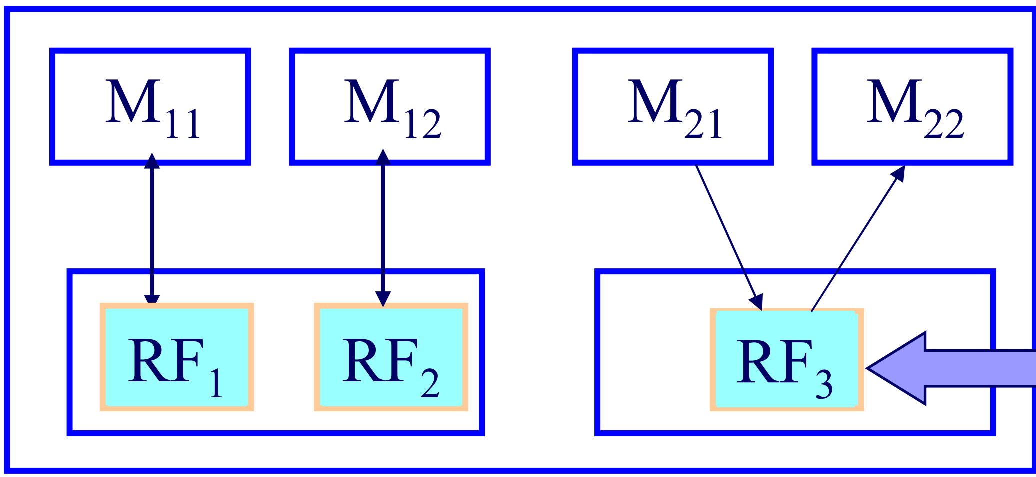
Phase 1: Enumerate FPGA buffers and FPGA unit function Designs

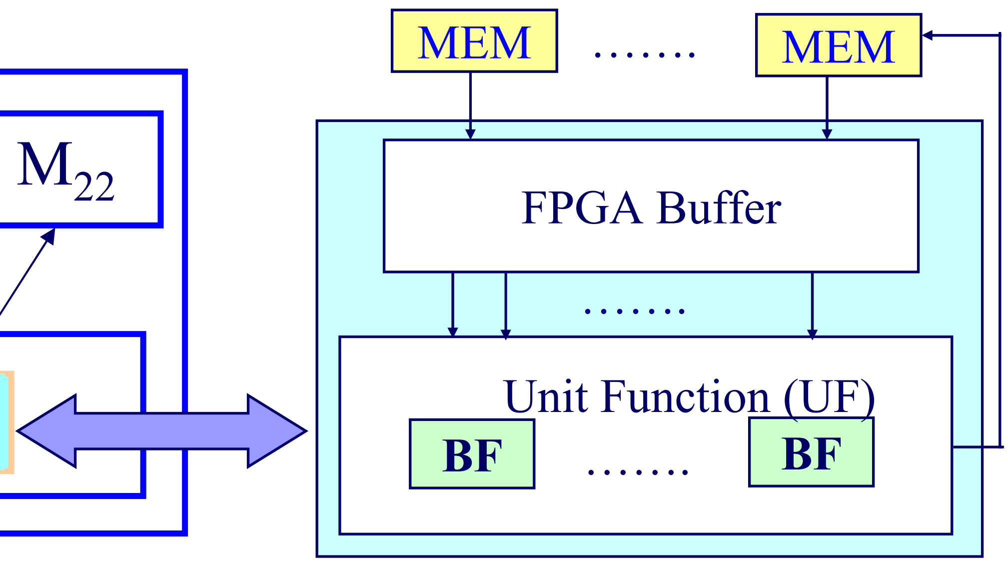




GTM Mapping Procedure (Cont.)

Phase 2: Select UF designs and FPGA buffers to compose RF designs. Select a group of RFs, bind them to FPGA chips and memory ports, and partition the workload among them so as to minimize the computation time.

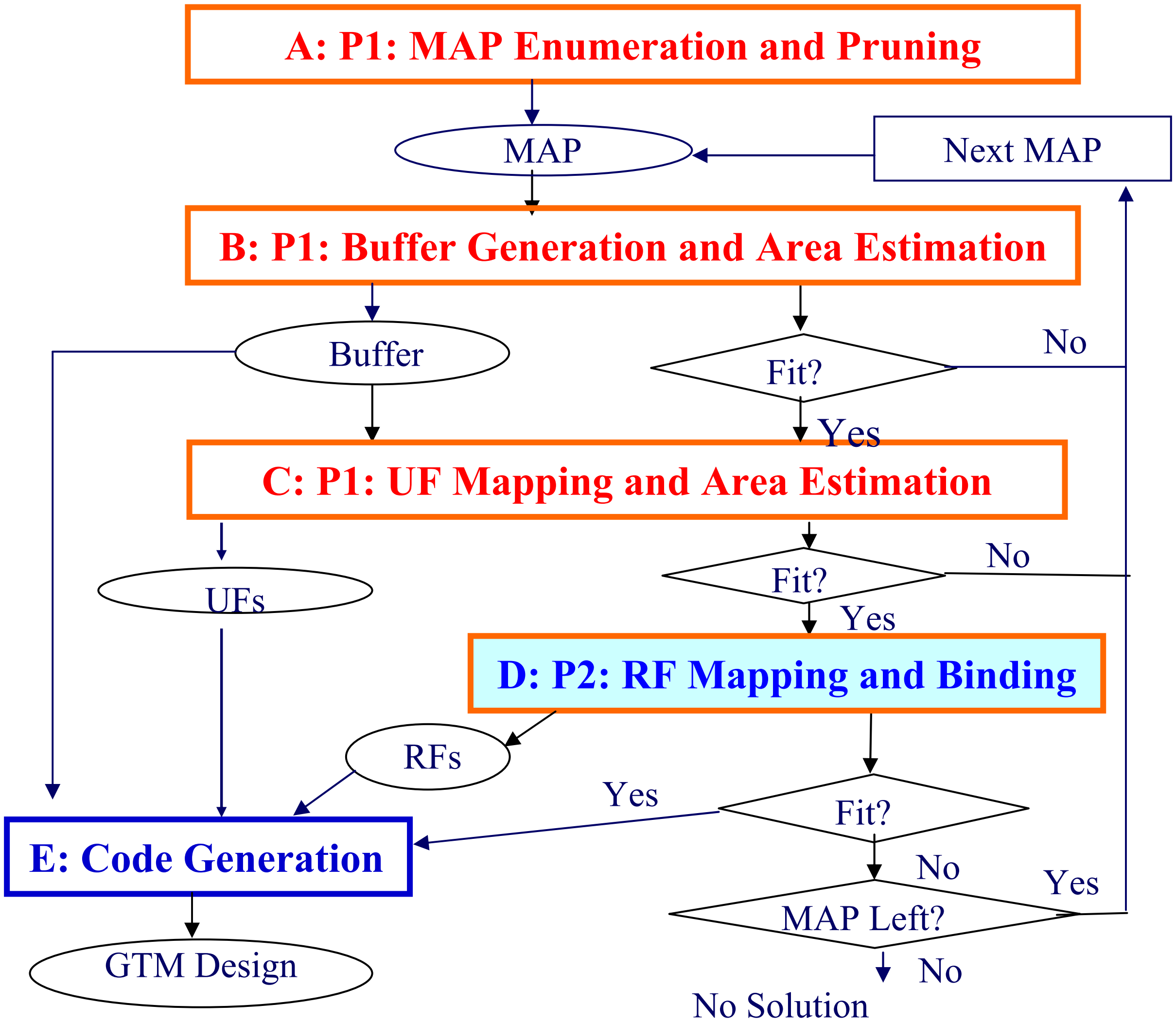




Region Function (RF)

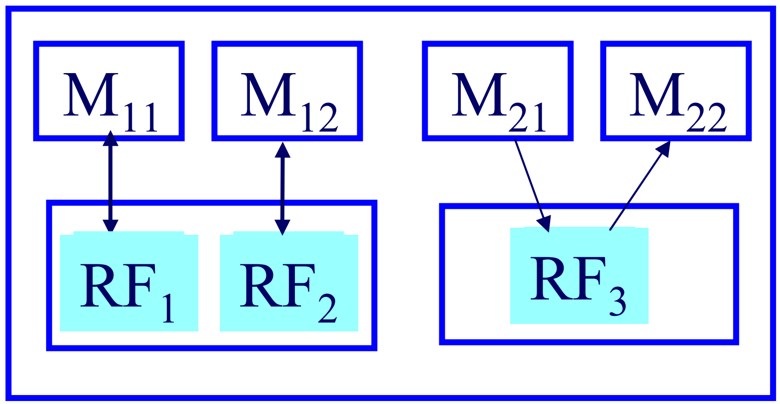
Each RF uses different memory ports and works on different image regions

RF Mapping and Binding



To minimize the computation time by

1. Selecting a group of RFs
2. Binding them to FPGA chips & memory ports
3. Partitioning the workload among them



RF Mapping and Binding (Cont.)

To minimize

$$\max\{\mathrm{Time}(\mathrm{RF}_{i,j}) \mid 1 \le i \le N_{FPGA}, \text{ and } 1 \le j \le q(i)\}$$

subject to

$$\begin{cases} \sum_{1 \le j \le q(i)} Area(RF_{i,j}) \le S_{FPGA},\ 1 \le i \le N_{FPGA} \\ \sum_{1 \le j \le q(i)} Port(RF_{i,j}) \le N_{MP},\ 1 \le i \le N_{FPGA} \\ \sum_{i=1}^{N_{FPGA}} \sum_{j=1}^{q(i)} WL(RF_{i,j}) = workload \end{cases}$$

$$workload = \sum_{s=1}^{r} d_s \times r_s$$

$$\mathrm{Time}(RF_{i,j}) = S(RF_{i,j}) \times WL(RF_{i,j})$$

$$S(RF) = (II / PF) \times CP \times C_{IMG}$$

RF Mapping and Binding (Cont.)

Reduce to one FPGA chip case
by partitioning the workload evenly among FPGA chips

$$\left.\begin{array}{l}\text{To minimize} \\ \qquad\qquad \max\{Time(\mathrm{RF_j}) \mid 1 \le j \le q\} \\ \text{subject to} \\ \qquad\qquad \begin{cases} \sum\limits_{1\le j\le q} Area(RF_j) \le S_{FPGA} \\ \sum\limits_{1\le j\le q} Port(RF_j) \le N_{MP} \\ Port(RF_j) \ge 1, 1 \le j \le q \\ \sum\limits_{j=1}^{q} WL(RF_j) = workload \end{cases}\end{array}\right\}$$

RF Mapping and Binding (Cont.)

Further reduce
by partitioning the workload "evenly" among RFs

$$\left.\begin{array}{l}\text{To maximize} \\ \qquad \displaystyle\sum_{1\le j\le q} 1/S(RF_j) \\ \text{subject to} \\ \qquad \begin{cases} \displaystyle\sum_{1\le j\le q} Area(RF_j) \le S_{FPGA} \\ \displaystyle\sum_{1\le j\le q} Port(RF_j) \le N_{MP} \end{cases}\end{array}\right\}$$

$$WL(j) = \frac{workload}{S(RF_j) \times (\sum_{1\le j\le q} 1/S(RF_j))}$$

RF Selection and Port Binding

To maximize

$$\sum_{1\le j\le q} 1/S(RF_j)$$

subject to

$$\begin{cases} \sum_{1\le j\le q} Area(RF_j) \le S_{FPGA} \\ \sum_{1\le j\le q} Port(RF_j) \le N_{MP} \end{cases}$$

Problem Size

$$N^1 + N^2 + \ldots + N^{N_{MP}} = (N^{N_{MP}} - 1) \times N/(N-1)$$

N is number of RF designs

For Naïve Approach

Assume N_{MP}=4, W_{PORT}=32, B_{DATA}=8, N_{MW}=1, α_W=1.0, R_{WIN}=9, and N_{AP}=20. Then N=114 and → 1.7×10^8

1. Naïve Approach
2. Further reduce to two sub-problems
   - Generalized Integer Partition Problem
   - Generalized Knapsack Problem

RF Selection and Port Binding (Cont.)

Generalized Integer Partition

$$i_1 + i_2 + \ldots + i_q \leq N_{MP}$$

$$1 \leq i_1 \leq i_2 \leq \ldots \leq i_q$$

$$1 \leq q$$

$$i_j = Port(RF_{i_j})$$

Generalized Knapsack Problem

To maximize

$$\sum_{1 \leq j \leq q} 1/\mathrm{S}(\mathrm{RF}_{i_j})$$

subject to

$$\sum_{1 \leq j \leq q} \mathrm{Area}(\mathrm{RF}_{i_j}) \leq S_{FPGA}$$

1. Solving GIP
2. For each solution of GIP, solving GKP
   A. Exhaustive Search, or
   B. Multi-Dimensional Binary Search (MDBS)

Integer Partition Problem

The solution to the problem is a set of *q*-vectors denoted by *S(q,n)*. For example, if *n=8* and *q=4*, then

$$i_1 + i_2 + \ldots + i_q = n$$

$$1 \le i_1 \le i_2 \le \ldots \le i_q$$

$$1 \le q \le n$$

$$S(q,n) = \{(1,1,1,5),(1,1,2,4),(1,1,33),(1,2,2,3),(2,2,2,2)\}$$

Define the an operator: Θ: I x (q-vectors)→(q+1)-vectors as

$$a\Theta(a_1, a_2, \ldots, a_q) = (a, a_1 + a - 1, a_2 + a - 1, \ldots, a_q + a - 1)$$

Then S(q,n) can be computed by the following recursive equation.

$$S(q,n) = \begin{cases} (n) & if \quad k = 1 \\ (1,1,\ldots,1) & if \quad k = n \\ \bigcup_{b=1}^{k} b\Theta S(q-1,(n-b)-(b-1)*(q-1)) & otherwise \end{cases}$$

Integer Partition Problem (Cont.)

A Dynamic Programming Algorithm to Compute *S(q,m)* *(1≤q≤m≤n)*

$$\text{For } \mathrm{i} = 1 \text{ to } \mathrm{n}$$
$$\quad \mathrm{S(i,i)} = (1,1,\ldots,1)$$
$$\text{For } \mathrm{j} = 2 \text{ to } \mathrm{n}$$
$$\quad \mathrm{S(1,j)} = (\mathrm{j})$$
$$\text{For } \mathrm{i} = 2 \text{ to } \mathrm{n-1}$$
$$\quad \text{For } \mathrm{j} = \mathrm{i}+1 \text{ to } \mathrm{n}$$
$$\quad\quad \mathrm{S(i,j)} = \Phi$$
$$\quad\quad \text{For } \mathrm{k} = 1 \text{ to } \lceil \mathrm{j/i} \rceil$$
$$\quad\quad\quad \mathrm{S(i,j)} = \mathrm{S(i,j)} \cup \mathrm{k}\Theta S(i-1,(j-k)-(k-1)\times(i-1))$$

	1	2			n-1	n
1	S(1,1)					
2	S(1,2)	S(2,2)				
n-1	S(1,n-1)	S(2,n-1)			S(n-1,n-1)	
n	S(1,n)	S(2,n)			S(n-1,n)	S(n,n)

MDBS: Preprocessing

Grouping: Each $RF_{i,j}$ in Cad(i) uses i memory ports

$$Cad(i) = \{RF_{i,j} \mid j = 1,2,\ldots,m(i)\},\ i = 1,2,\ldots N_{MP}$$

Sorting:

$$S(RF_{i,1}) < S(RF_{i,2}) < \ldots < S(RF_{i,m(i)}),\ i = 1,2,\ldots N_{MP}$$

Assumption:

$$Area(RF_{i,1}) > Area(RF_{i,2}) > \ldots > Area(RF_{i,m(i)}), i = 1,2,\ldots N_{MP}$$

MDBS: Algorithm

For a solution to $(i_1, i_2, \ldots, i_q)$ GIP, compute

sum of the largest RF areas (*SLA*) $$SLA = \sum_{1 \le j \le q} Area(RF_{i_j,1})$$

sum of the smallest RF areas (*SSA*) $$SSA = \sum_{1 \le j \le q} Area(RF_{i_j,m(j)})$$

If $SLA < S_{FPGA}$, $\{RF_{i_j,1} \mid 1 \le j \le q\}$ is a feasible solution.

If $SSA > S_{FPGA}$, there is no solution.

Otherwise, compute sum of the median RF areas (*SMA*)

$$SMA = \sum_{1 \le j \le q} Area(RF_{i_j,m(j)/2})$$

MDBS: Algorithm (Cont.)

If $SMA > S_{FPGA}$, then $\mathrm{Cad}(i_j)$ ($1 \le j \le q$) can be divided into two parts:

$Cad(i_j,0) = \{RF_{i_j,k} \mid k = 1,2,\ldots,m(i)/2\}$ $Cad(i_j,1) = \{RF_{i_j,k} \mid k = m(i_j)/2+1,\ldots,m(i_j)\}$

Picking q RF from these $2\times q$ sets gives 2^q combinations. Then the searching problem can be broken up as 2^q-1 sub-problems with these combinations except one combination $\mathrm{Cad}(i_1,0),\ldots, \mathrm{Cad}(i_q,0)$.

If $SMA \le SFPGA$, then $\mathrm{Cad}(i_j)$ ($1 \le j \le q$) can be divided into two parts:

$Cad(i_j,0) = \{RF_{i_j,k} \mid k = 1,2,\ldots,m(i)/2-1\}$ $Cad(i_j,1) = \{RF_{i_j,k} \mid k = m(i_j)/2,\ldots,m(i_j)\}$

Picking q RF from these $2\times q$ sets gives 2^q combinations. Then the searching problem can be broken up as 2^q-1 sub-problems with these combinations except one combination $\mathrm{Cad}(i_1,1),\ldots, \mathrm{Cad}(i_q,1)$.

MDBS: Example

Assume S_{FPGA}=37 and (1, 2) is a solution to GIP.
Assume there are 6 RFs in Cad(1) and 5 RFs in Cad(2)
Their areas are listed in the following table.

j	1	2	3	4	5	6
Area($RF_{1,j}$)	24	20	16	14	13	12
Area($RF_{2,j}$)	35	29	26	22	18	NA

$SLA = 24+35 = 59 > S_{FPGA}$ and SSA=12+18 = 30 < S_{FPGA}.
As $SMA = 16+26 = 42 > S_{FPGA}$, three sub-problems are needed to consider.

MDBS: Example (Cont.)

Sub(1): j	1	2	3	4	5	6
Area($RF_{1,j}$)	NA	NA	NA	14	13	12
Area($RF_{2,j}$)	NA	NA	NA	22	18	NA

For sub-problem (1), as SLA = 14+22 = 36 < S_{FPGA}, ($RF_{1,4}$, $RF_{2,4}$) is a feasible solution

Sub(2): j	1	2	3	4	5	6
Area($RF_{1,j}$)	NA	NA	NA	14	13	12
Area($RF_{2,j}$)	35	29	26	NA	NA	NA

For sub-problem (2), as SSA = 12+26 = 38 > S_{FPGA}, there is no solution

Sub(3): j	1	2	3	4	5	6
Area($RF_{1,j}$)	24	20	16	NA	NA	NA
Area($RF_{2,j}$)	NA	NA	NA	22	18	NA

For sub-problem (3), SLA = 46 and SSA = 34. As $SMA = 42 > S_{FPGA}$, it splits into three sub-problems again

MDBS: Example (Cont.)

Sub(3-1): J	1	2	3	4	5	6
Area($RF_{1,j}$)	NA	NA	16	NA	NA	NA
Area($RF_{2,j}$)	NA	NA	NA	NA	18	NA

Sub-problem (3-1) provides another feasible solution ($RF_{1,3}$, $RF_{2,5}$),

Sub(3-2): J	1	2	3	4	5	6
Area($RF_{1,j}$)	NA	NA	16	NA	NA	NA
Area($RF_{2,j}$)	NA	NA	NA	22	NA	NA

Sub-problems (3-2) has no solution.

Sub(3-3): J	1	2	3	4	5	6
Area($RF_{1,j}$)	24	20	NA	NA	NA	NA
Area($RF_{2,j}$)	NA	NA	NA	NA	18	NA

Sub-problems (3-3) has no solution.

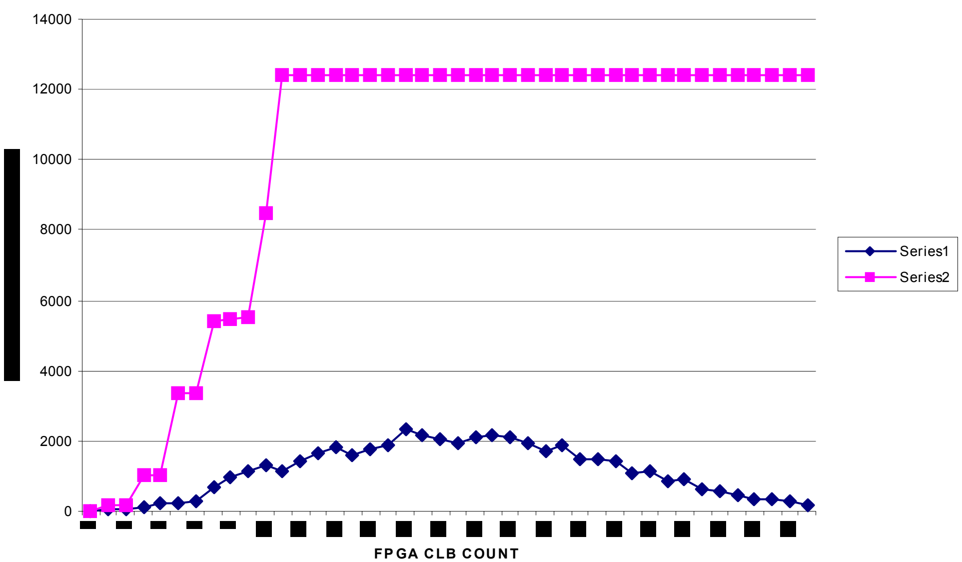
The final solution for (1, 2) then will be one of the two feasible solutions that has maximum sum of speeds.

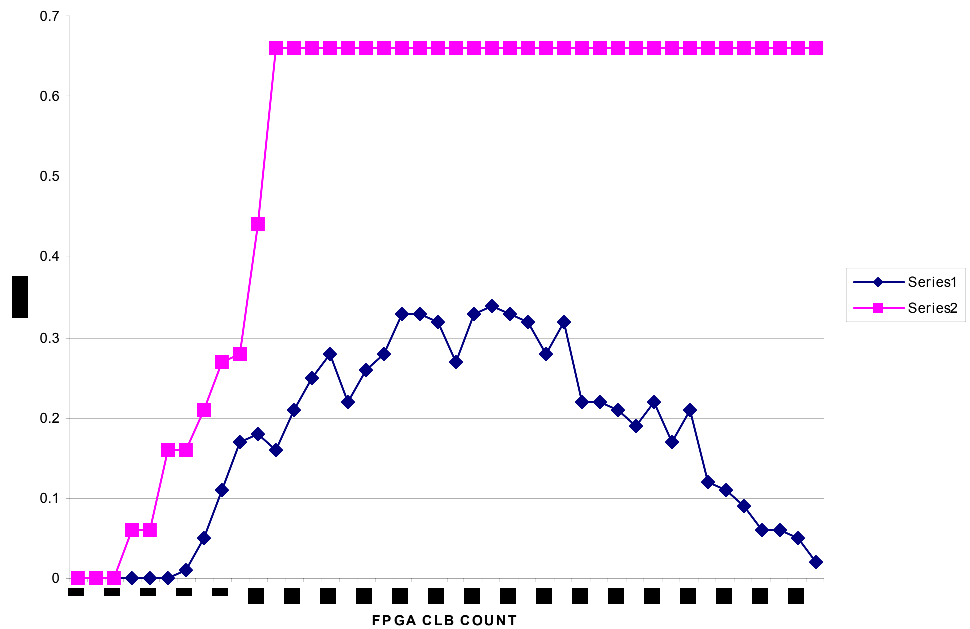
Efficiency of MDBS

	Complexity Assume m(i)=m, N_{MP}=n	Example: $N_{MP}=4$, $W_{PORT}=32$, $B_{DATA}=8$, $N_{MW}=1$, $\alpha_W=1.0$, $R_{WIN}=3$ and $N_{AP}=9$	
		Space Size	Time(in Second)
Naïve	$m^n \times n^n$	2,880,952	148.05
Exhaustive Search	m^n	9,982	0.526
Multi-Dimensional Binary Search	$\begin{cases} \theta(m^{\log_2(2^n-1)}) & \text{if } n>1 \\ \theta(\log_2 m) & \text{if } n=1 \end{cases}$	1,138	0.1288

Comparison	Space	Time
Naïve / Exhaustive Search	2880952/9982=288	148.05/0.526=281
Exhaustive Search / MDBS	9982/1138=9	0.526/0.1288=3

Efficiency of MDBS (Cont.)





Series 1: Multi-Dimensional Binary Search
Series 2: Exhaustive Search