BAMSE: A Balanced Mapping Space Exploration Algorithm for GALS-based Manycore Platforms

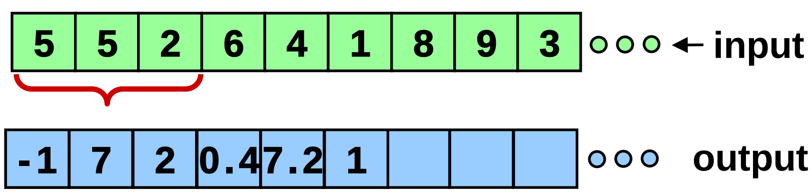
Mohammad H Foroozannejad
Brent Bohnenstiehl
Soheil Ghiasi

Department of Electrical & Computer Engineering
University of California, Davis

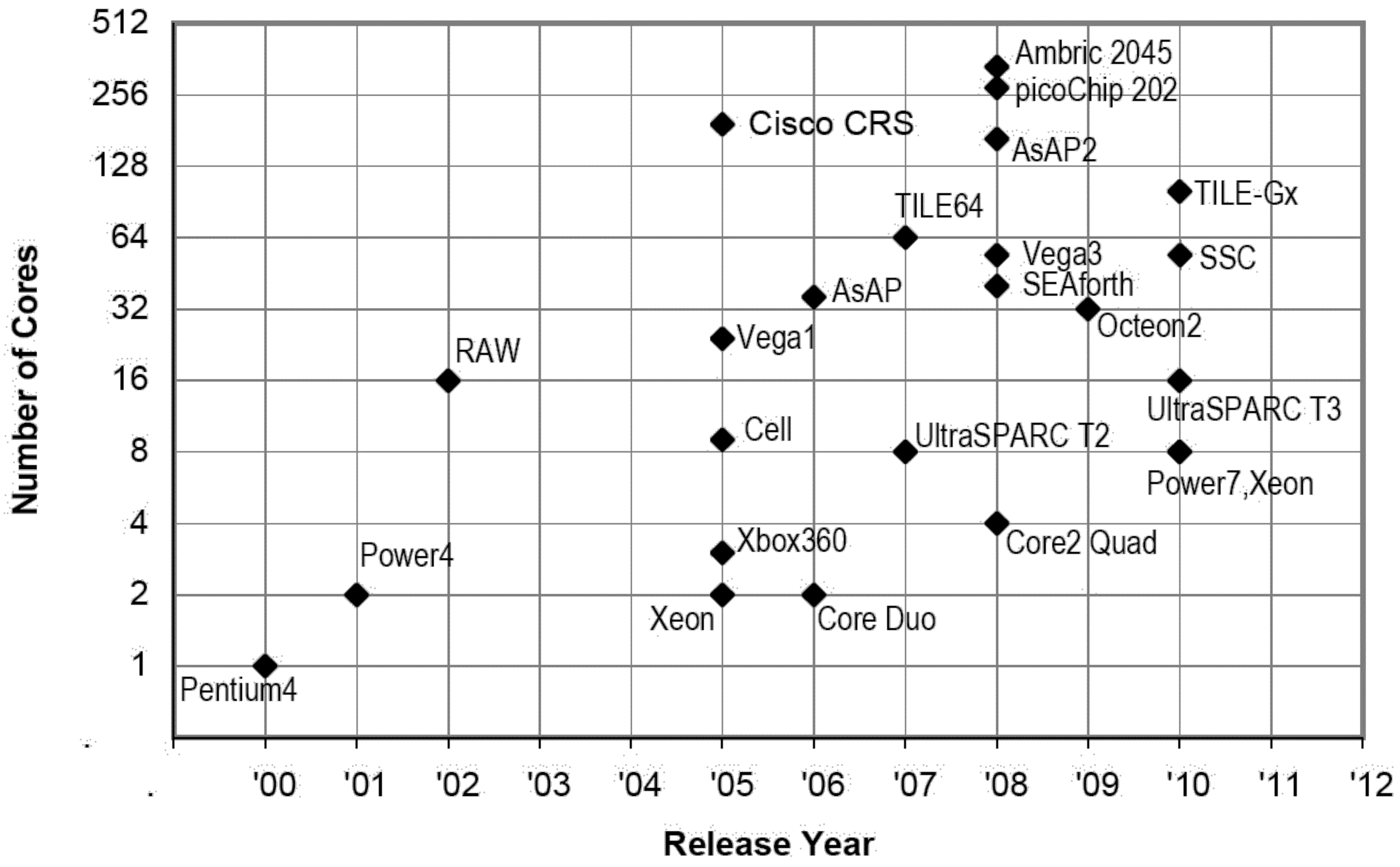
Target Applications

- Streaming
  - Cell phones , mp3 players, video conference,
    data encryption, graphics, packet inspection, imaging, cellular base stations
- Properties
  - Infinite sequence of data items
  - At any given time, operates on a small window of this sequence

```
//53° around the z axis
const R[3][3]={
      {0.6,-0.8, 0.0},
      {0.8, 0.6, 0.0},
      {0.0, 0.0, 1.0}}
Rotation3D {
 for (i=0; i<3; i++)
  for (j=0; j<3; j++)
   B[i] += R[i][j] * A[j]
 }
```



Trend in Processor Architecture



[Hashemi'11]

Productive Programming of Many-Core Platforms

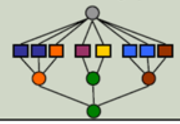
Mohammad H. Foroozannejad, Matin Hashemi, Trevor Hodges, Soheil Ghiasi
Electrical and Computer Engineering Department
University of California, Davis
http://leps.ece.ucdavis.edu

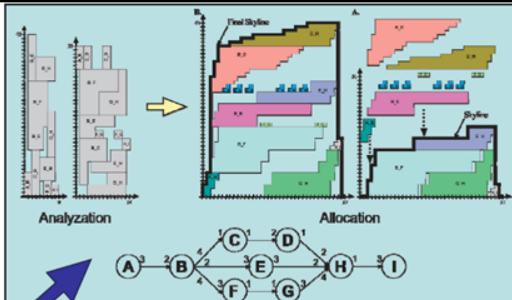
Formless: Application Parameter-Space Exploration [1]

The application is defined as a set of "parameterized" actors and their connections. The tool automatically synthesizes a task graph that best fits the target hardware platform.

Task Graph Partitioning [2]

The goal is to partition the task graph among processors. For example, in the graph below, we have 7 partitions marked by colors. As the primary goal, the algorithm tries to find the partition that results in the maximum overall throughput.

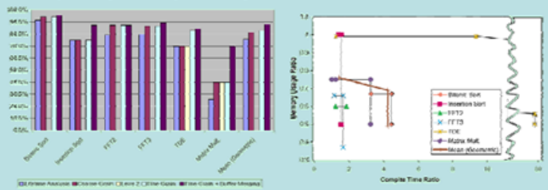




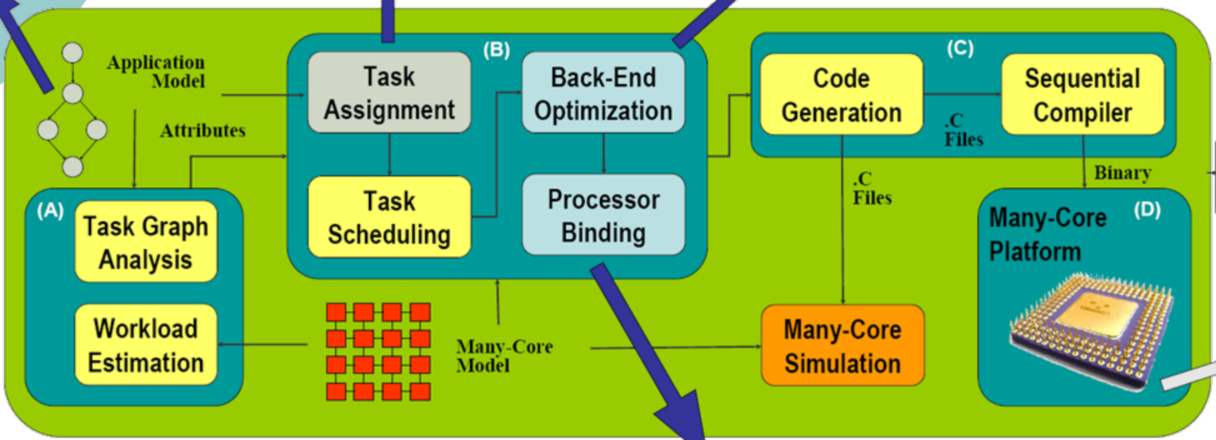
Memory Optimization [3]

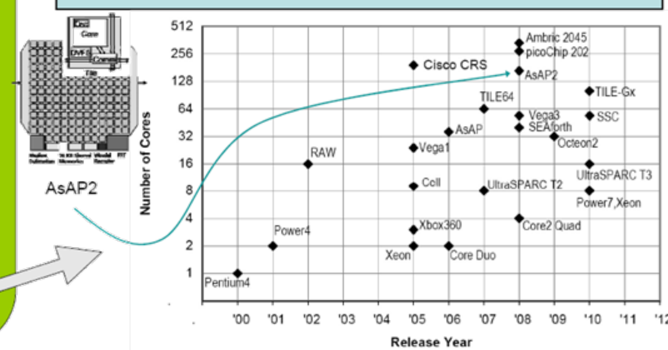
- Define the desired level of granularity for the analysis
- Analyze channel buffer requirements at the defined level
- Allocate the buffers in the shared memory space using genetic algorithm

Memory Saving Results [LCTES'10]:



- Left: Total savings compared to base-line
- Right: Trade off between optimality and complexity compared to lifetime analysis





List of Latest Publications

[1] Matin Hashemi, Mohammad H. Foroozannejad, Christoph Etzel, Soheil Ghiasi, "FORMLESS: Scalable Utilization of Embedded Manycores in Streaming Applications", to appear in Proceedings of the ACM SIGPLAN/SIGBED 2012 Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES), Beijing, China

[2] Matin Hashemi, Soheil Ghiasi, "Versatile Task Assignment for Heterogeneous Soft Dual-Processor Platforms", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), Vol 29, No. 3, pp. 414 - 425, 201

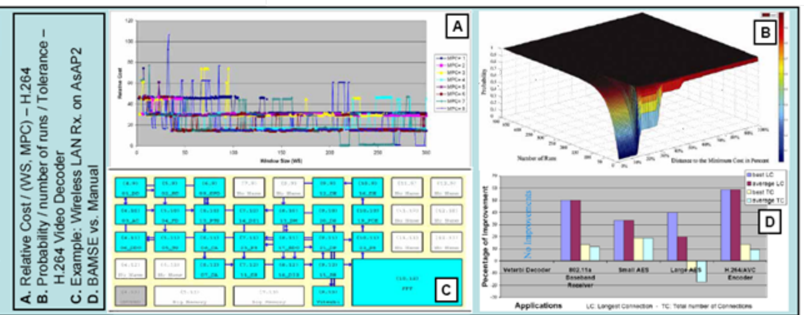
[3] Mohammad H. Foroozannejad, Trevor Hodges, Matin Hashemi, Soheil Ghiasi, "Post-Scheduling Buffer Management Tradeoffs in Synthesis of Streaming Applications", to appear in ACM Transactions on Design Automation of Electronic Systems (TODAES).

[4] Mohammad H. Foroozannejad, Brent Bohnenstiehl, Soheil Ghiasi, "BAMSE: A Balanced Mapping Space Exploration Algorithm for Manycore Processors".

BAMSE: Balance Mapping Space Exploration [4]

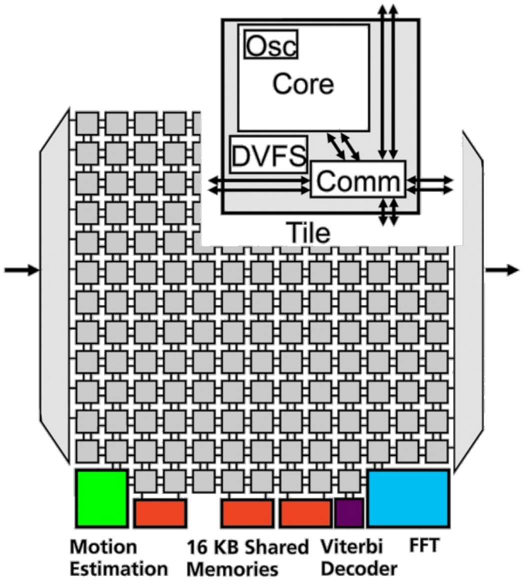
- Input: Virtual Processor Graph
  Nodes and edges represent processors and channels, respectively
- Objective: Mapping of virtual to physical resources
- Optimization Criteria:
  1. Minimize the longest communication link
  2. Minimize total communication distances
  3. Area
  4. Other application-dependent criteria

A. Relative Cost / (WS, MPC) – H.264
B. Probability / number of runs / Tolerance – H.264 Video Decoder
C. Example: Wireless LAN Rx. on AsAP2
D. BAMSE vs. Manual



Motivating Platform

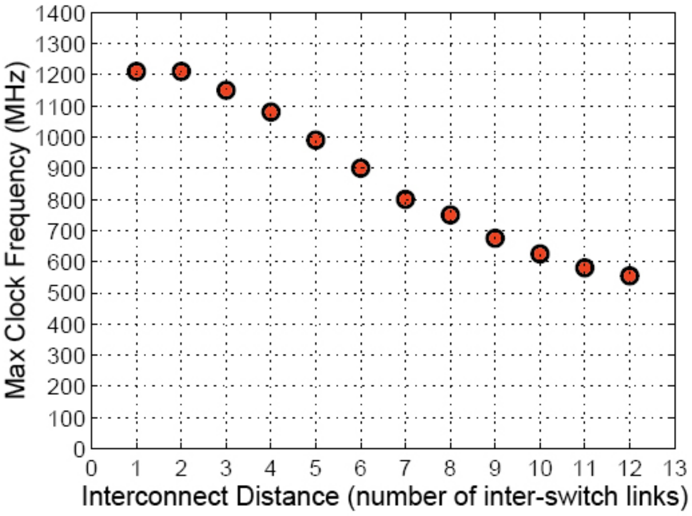
- Key Features
  - 164 Enhanced Programmable Processors
  - 3 Dedicated-purpose processors
  - 3 Shared memories
  - Long-distance circuit-switched communication network
  - Dynamic Voltage and Frequency Scaling (DVFS)



[Baas et al.'08]

Globally-Synchronous Locally-Asynchronous (GALS) Architecture

- The same clock used to supply the source processor is used for the communication
  - Long communication slows down the source processor regardless of the communication volume
- Static Link Allocation (limited resources)



[Baas et al.'10]

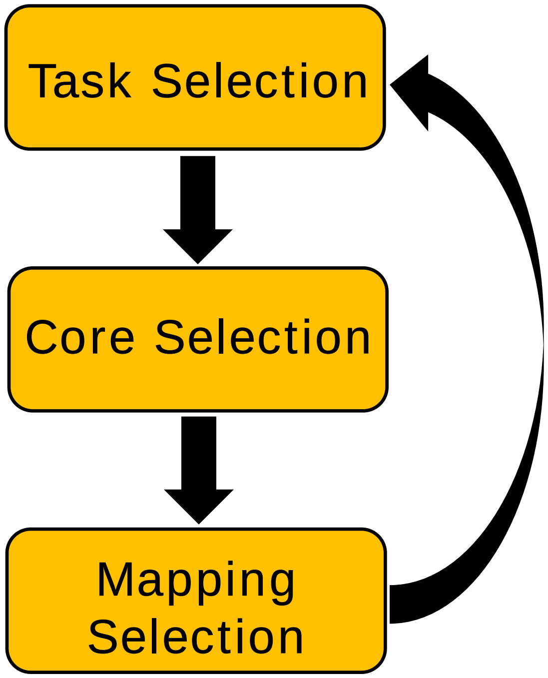
Problem Statement

- Task graph *G* in which, the vertices model application tasks, and edges represent inter-task communication.
- The hardware graph *H* consists the set of available cores on the chip connected, and L, a subset of C×C representing inter-core links
- <u>Objective: An embedding of the task graph on the hardware graph</u>
  - Improved application performance and energy dissipation
  - Graceful runtime-quality tradeoff (applicable to dynamic mapping)

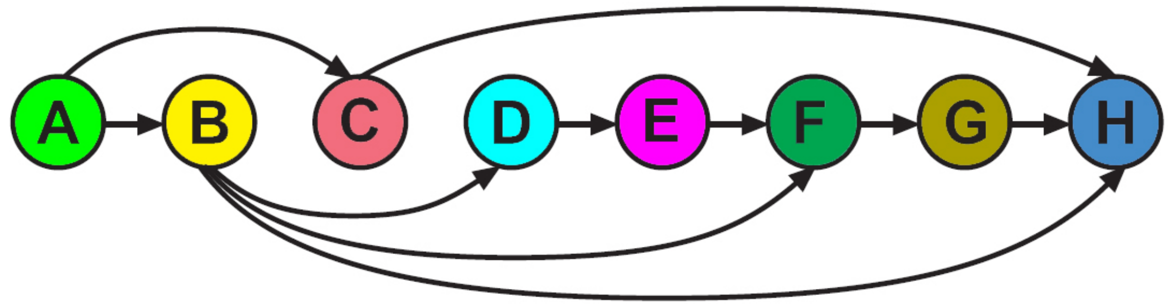
BAMSE Overview

- Constructive Approach
- Task Selection
  - Tasks visited and handled in some order
- Core Selection
  - Candidate cores for allocating the task
  - Generate partial mappings and add to a queue
- Mapping Selection
  - Maintain a number of promising partial mappings
  - Avoid state explosion

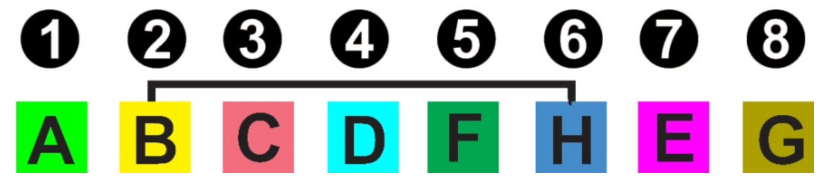
- Balancing greediness (runtime) with mapping space coverage (quality) using a few parameters
- Priority-based multi objective cost function:
  - Longest Connection (LC)
  - Total number of Connections (TC)
  - Cores Bounding Box Area (A)



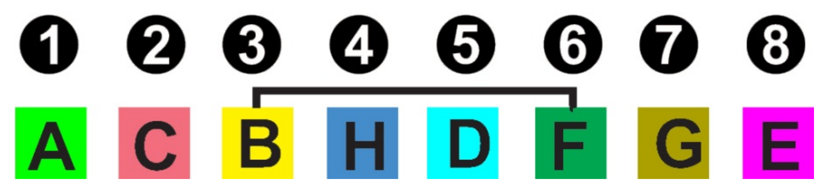
Task Selection



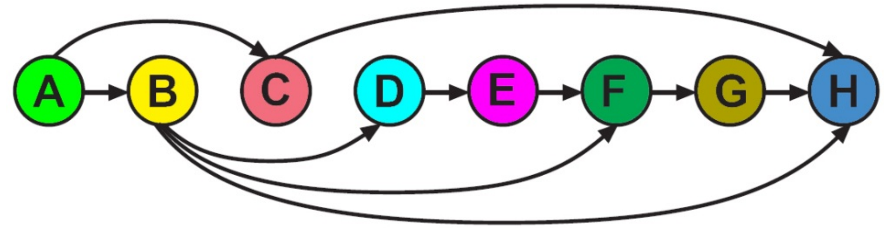
- Breath First Search
  - Unconstrained BFS



    - Maximum Distance to Children (MDC) = 4
  - Cuthill-McKee BFS
    - Children are sorted in increasing order of their degree (MDC = 3)



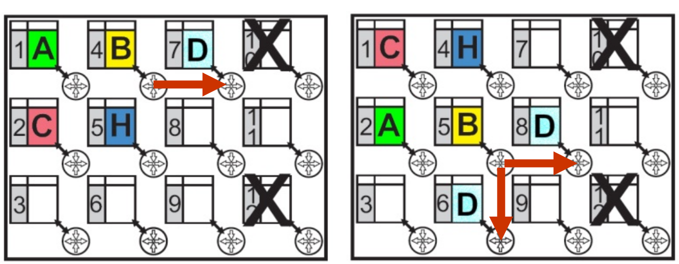
Core Selection



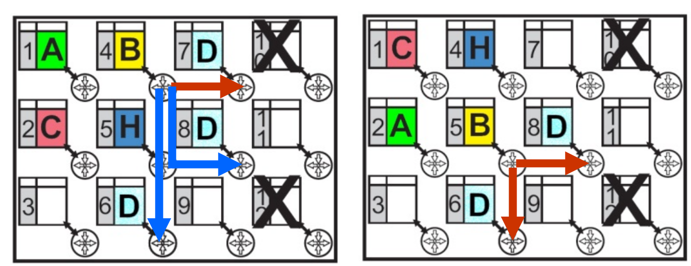
1	2	3	4	5	6	7	8
A	C	B	H	D	F	G	E

- Select cores that are close to the mapped connected tasks
  - Intuition: minimize the cost increase
  - Available cores are considered in batches, according to their contribution to the cost function
    - Parameter: Minimum number of Potential Cores(MPC)
    - Unavailable cores are removed from consideration

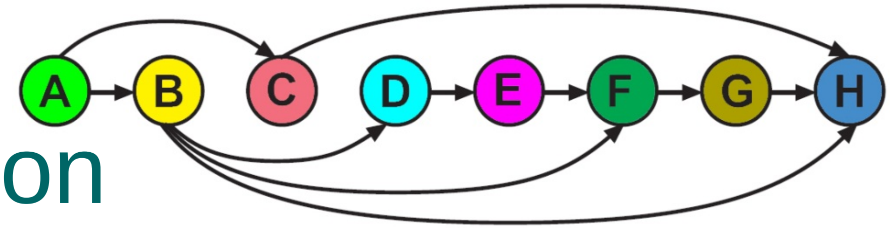
MPC = 1



MPC = 2

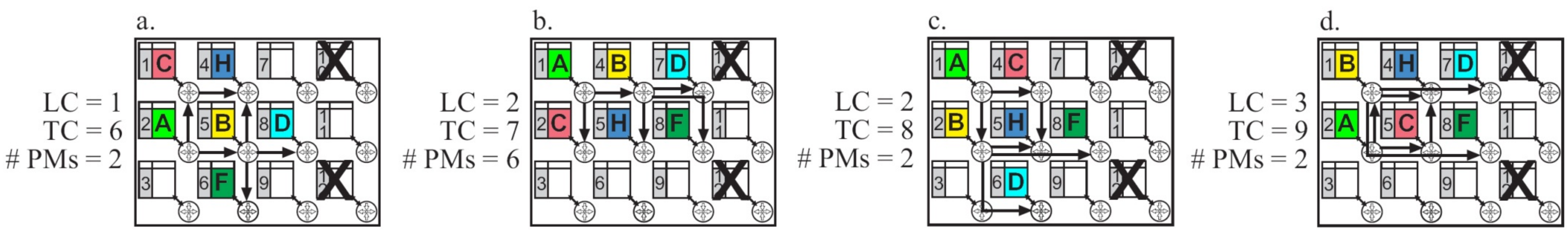


Mapping Selection



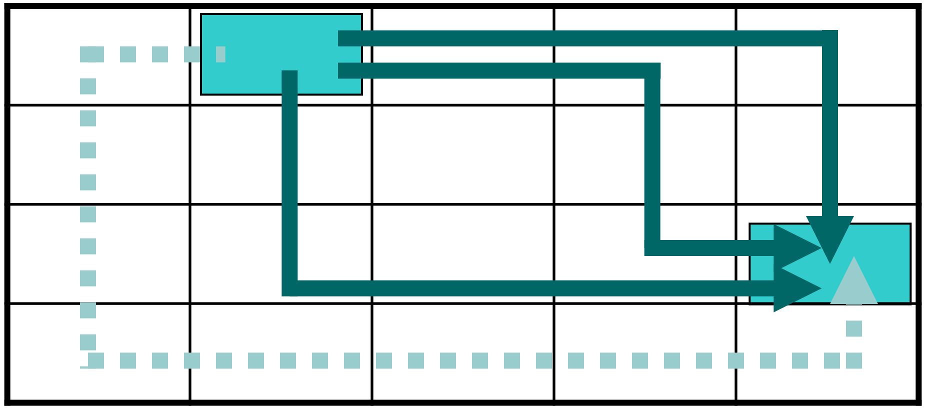
- Generated partial mappings are added to a sorted list (based on cost)
- To avoid state explosion, the list is trimmed
  - <u>Parameter:</u> Window Size (WS)

The following Partial Mappings are created after mapping node F
There are 12 mappings in the list with four different costs. An example partial mapping for each cost is shown.



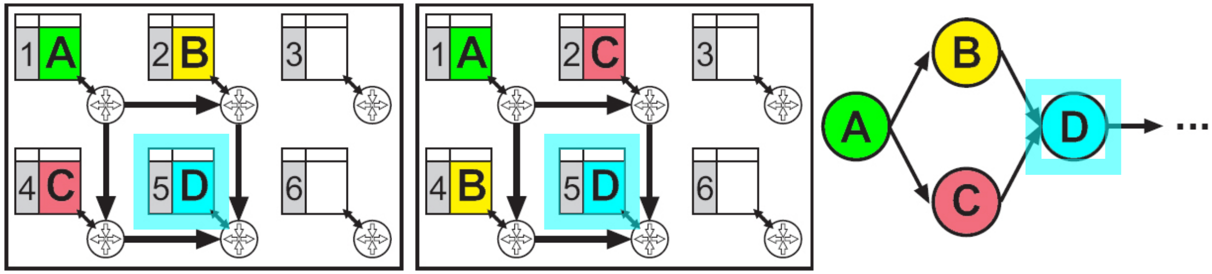
Link Assignment

- Due to limited network resources, not all mappings yield feasible implementations.
- Simultaneous mapping and link assignment
  - A bookkeeping table keeps track of reserved interconnect resources.



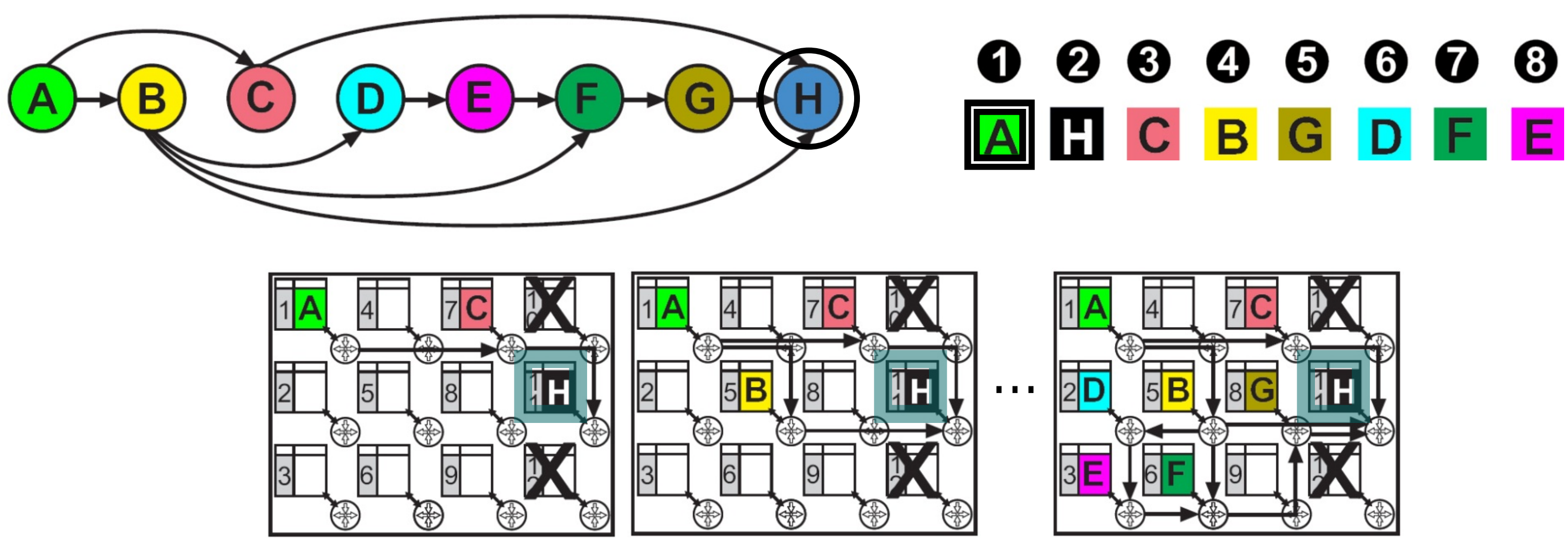
Enhancements to the Baseline

- Look-Ahead
  - Mapping some ‘future’ tasks to better sort the partial mapping list.
  - Helps to reduce the Window Size
  - Parameter: The Forwarding Number (FN). MDC can be heuristically used as FN to estimate the impact of all children of visited tasks.
- Redundant Mapping Elimination
  - Based on mapping of tasks with connection to unmapped tasks, and the cost of partial mappings



Fixed Mappings

- Fixed mappings are dictated by the platform architecture (e.g., hardware accelerators) or programmers preference/insight
  - Handled naturally by prioritizing their ordering in Task Selection

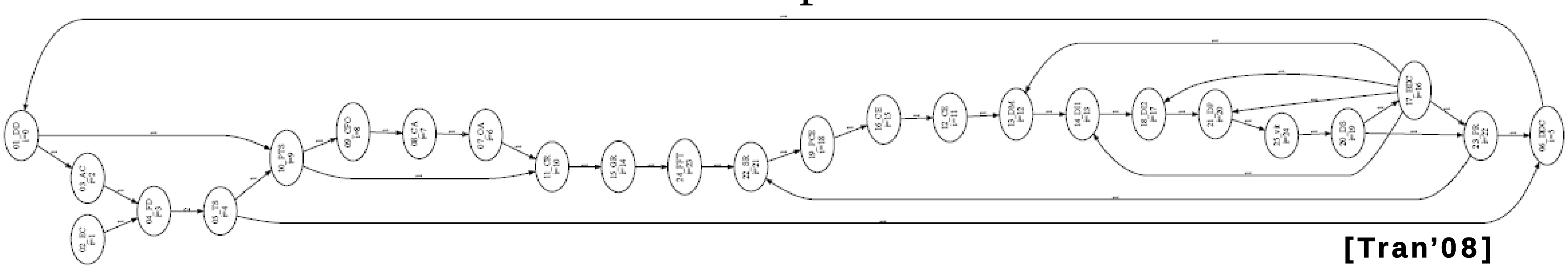


Empirical Validation

Application Name	# Nodes	# Edges	D	MDC
Viterbi Decoder	30	35	3	4
802.11a B.B. Rx.	25	40	6	9
Small AES	59	79	3	4
Large AES	137	176	6	8
H.264/AVC Encoder	115	165	7	24

D: Maximum undirected degree of the task graph
MDC: Maximum Distance to Children with Cuthill-McKee BFS

802.11a Broad Band Receiver Graph

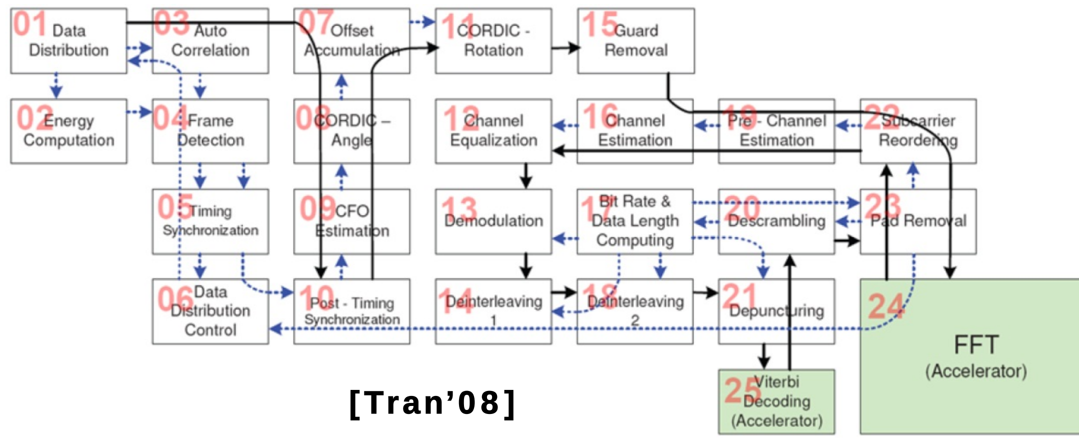


[Tran'08]

Example: 802.11a Receiver

Manual Mapping

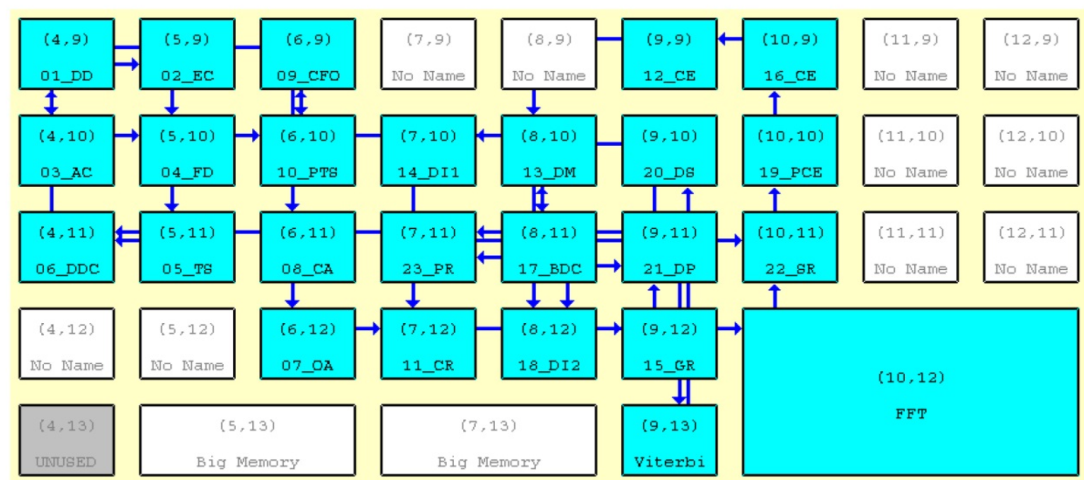
Longest Connection = 6
Total Connections = 59



[Tran'08]

BAMSE

Longest Connection = 3
Total Connection = 51



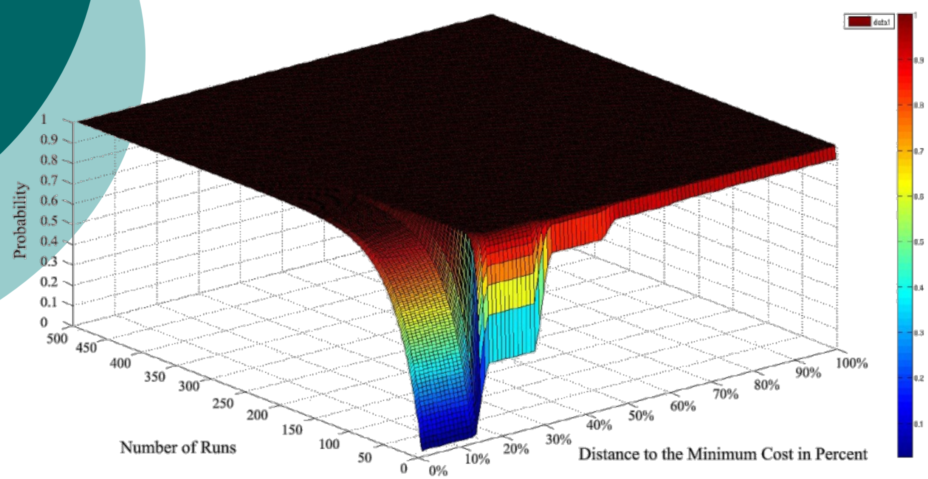
Empirical Validation

Application		LC	TC	Time
Viterbi Decoder	Manual	1	35	-
	BAMSE	1	35	1 (sec)
	ILP**	1	35	46 (hours)
802.11a B.B. Rx.	Manual	6	58	-
	BAMSE	3	51	13 (sec)
	ILP**	3	51	58 (hours)
Small AES	Manual	3	106	-
	BAMSE	2	86	2 (sec)
	ILP*	3	105	10 (days)
Large AES	Manual	5	254	-
	BAMSE	3	273	170 (sec)
	ILP*	5	328	10 (days)
H.264/AVC Encoder	Manual	17	353	-
	BAMSE	6	336	273 (sec)
	ILP*	7	288	10 (days)

- ILP* number are obtained by terminating the solver after 10 days.
- ILP** are optimal, however, a smaller hardware graph (Mesh of 6X6 cores) is exposed to the solver to accelerate it.

Parameter Space Exploration

$$Rel\_Cost^{app}_{(ws,mpc)} = \frac{cost^{app}_{(ws,mpc)}}{mincost^{app}} - 1$$



Based on random sampling of
the parameter space

Acceptance Threshold of Relative Cost:

0% (best result) → LC = 6, TC = 336

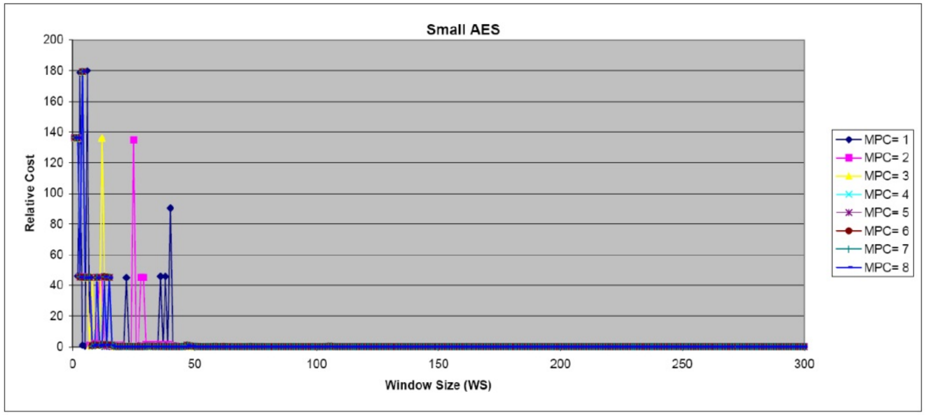
# of acceptable mappings = 4

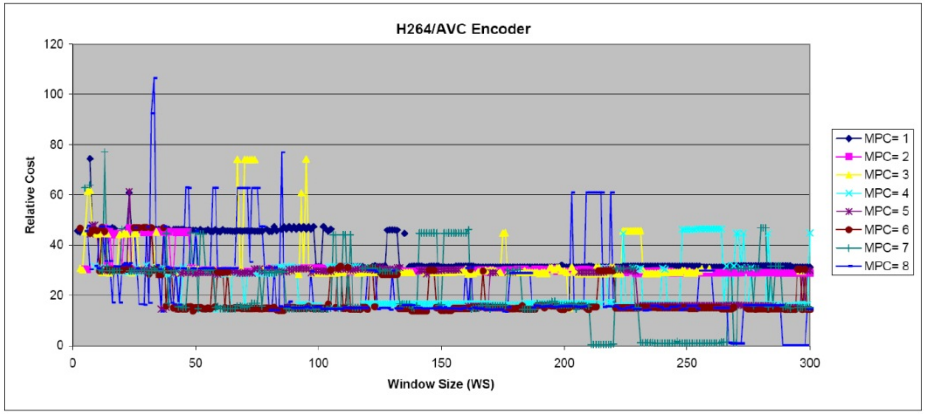
10% → LC = 6, TC = 356

# of acceptable mappings = 10

50% → LC = 9, TC = 456

# of acceptable mappings = 2150





Data from 2400 runs of
WS = 1 to 300 and *MPC*= 1 to 8

Future Work

- Automatic Parameter Tuning
  - Space too large for manual configuration
- Core-Task “suitability metric”:
  - Matching tasks with intensive workload to faster processors
- Dynamic Mapping
  - Launching and terminating applications
  - Incremental mapping

Questions?

Thank you