

Automatic Architecture Selection for Custom Processors

Jelena Trajkovic and Daniel D. Gajski

{jelenat, gajski}@ics.uci.edu

Center for Embedded Computer Systems
University of California, Irvine

SRC Research Task ID: 1118.001

Task Leader: Daniel D. Gajski

Paper number: 9.5



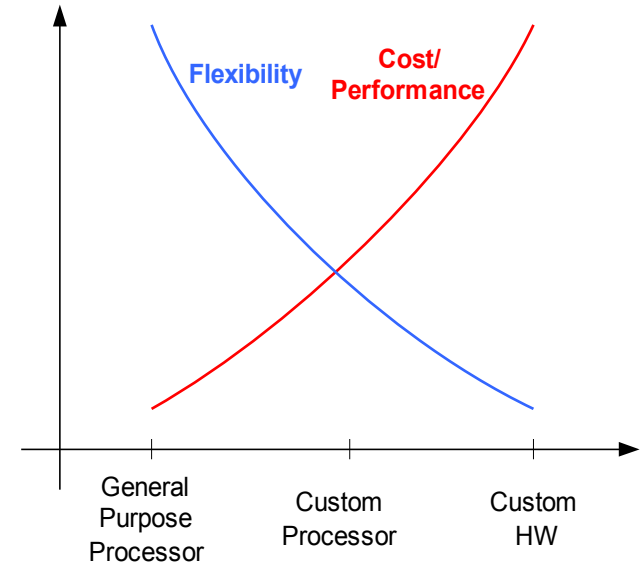
Outline

- **Motivation for automatic datapath generation**
- **Custom processor design using automatic datapath generation**
- **Datapath optimization techniques**
- **Experimental results**
- **Conclusion and future directions**



Motivation

- **Need for Custom Processors (CP)**
 - Rising complexity of SoC designs
 - Short time to market, performance goals, cost constraints
- **CP features**
 - Datapath optimized for application domain
 - Programmable control
 - What is the best datapath to execute given C code?
- **Proposed CP design**
 - Datapath is extracted from application C code
 - Able to handle ~10000 lines of code (LoC)
 - Datapath design is controllable
 - Control logic is generated for selected datapath



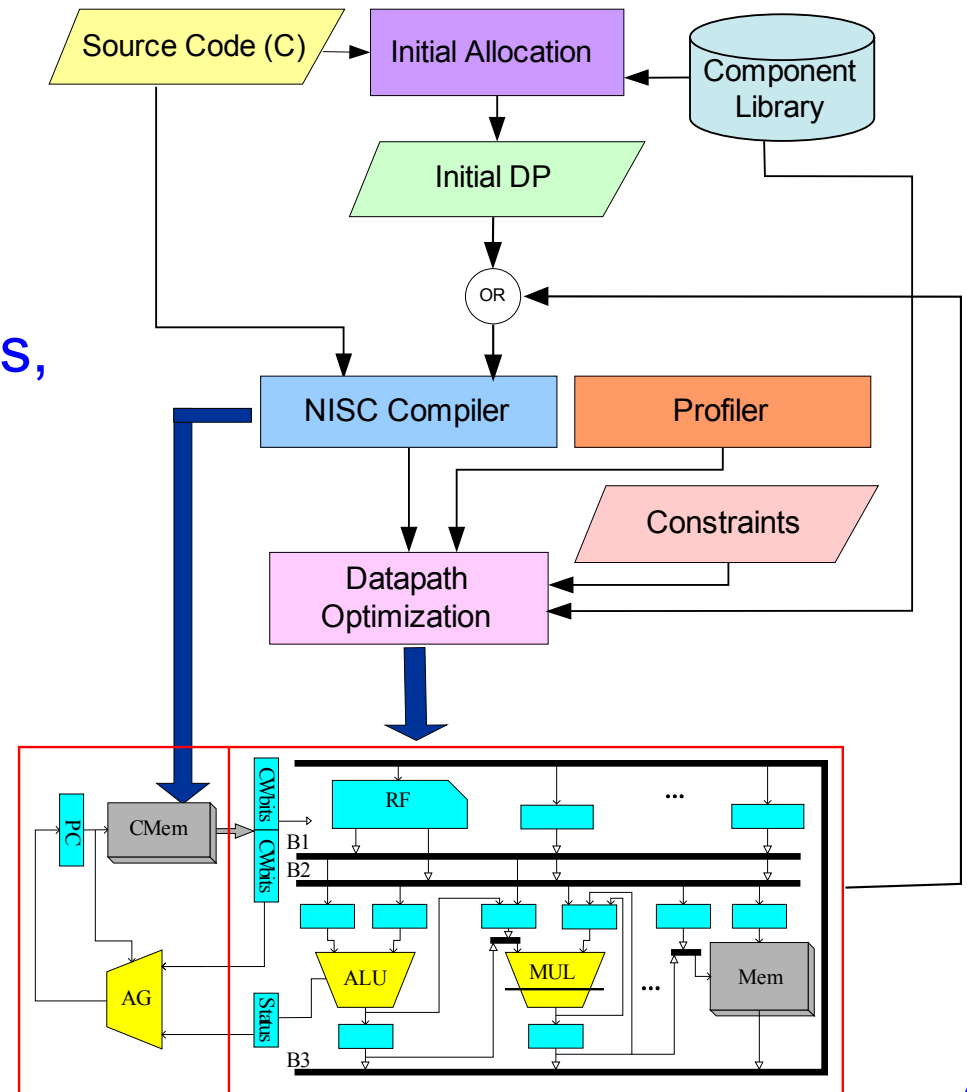
Custom Processor Design

2. Initial Datapath (DP)

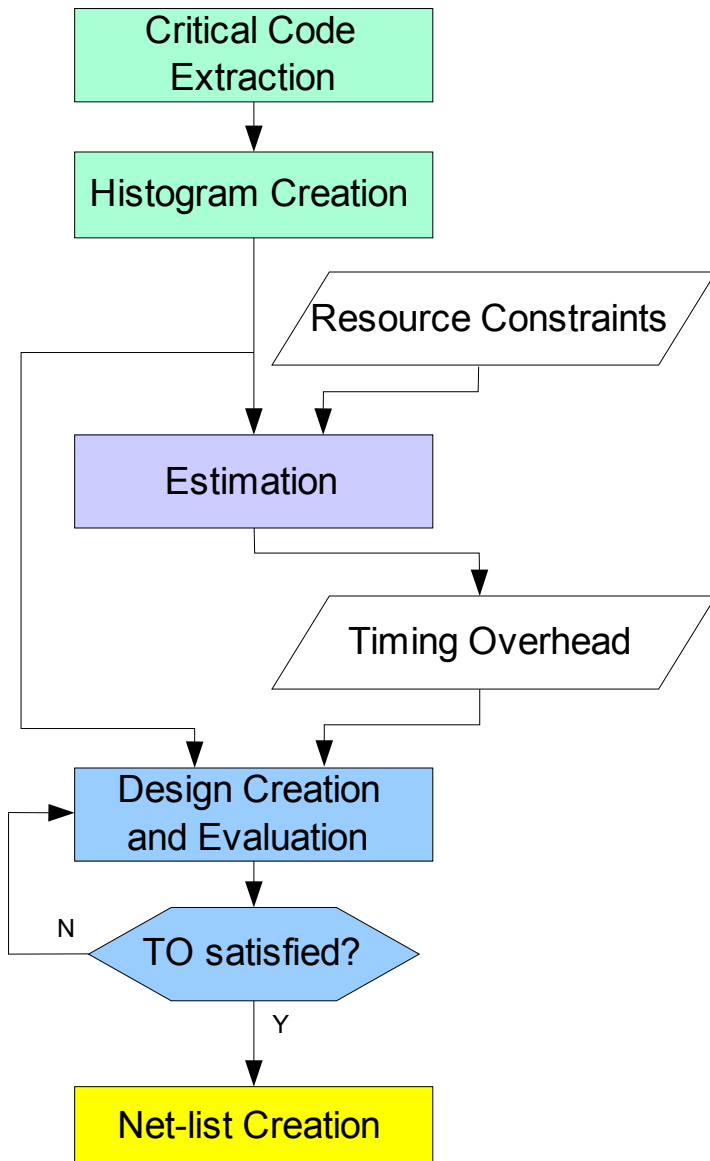
- Minimum execution time
- Constrained by available components (functional units, storage units, buses...)

4. Datapath Optimization

- Critical code selection
- Intermediate DP generation
- DP evaluation and refinement



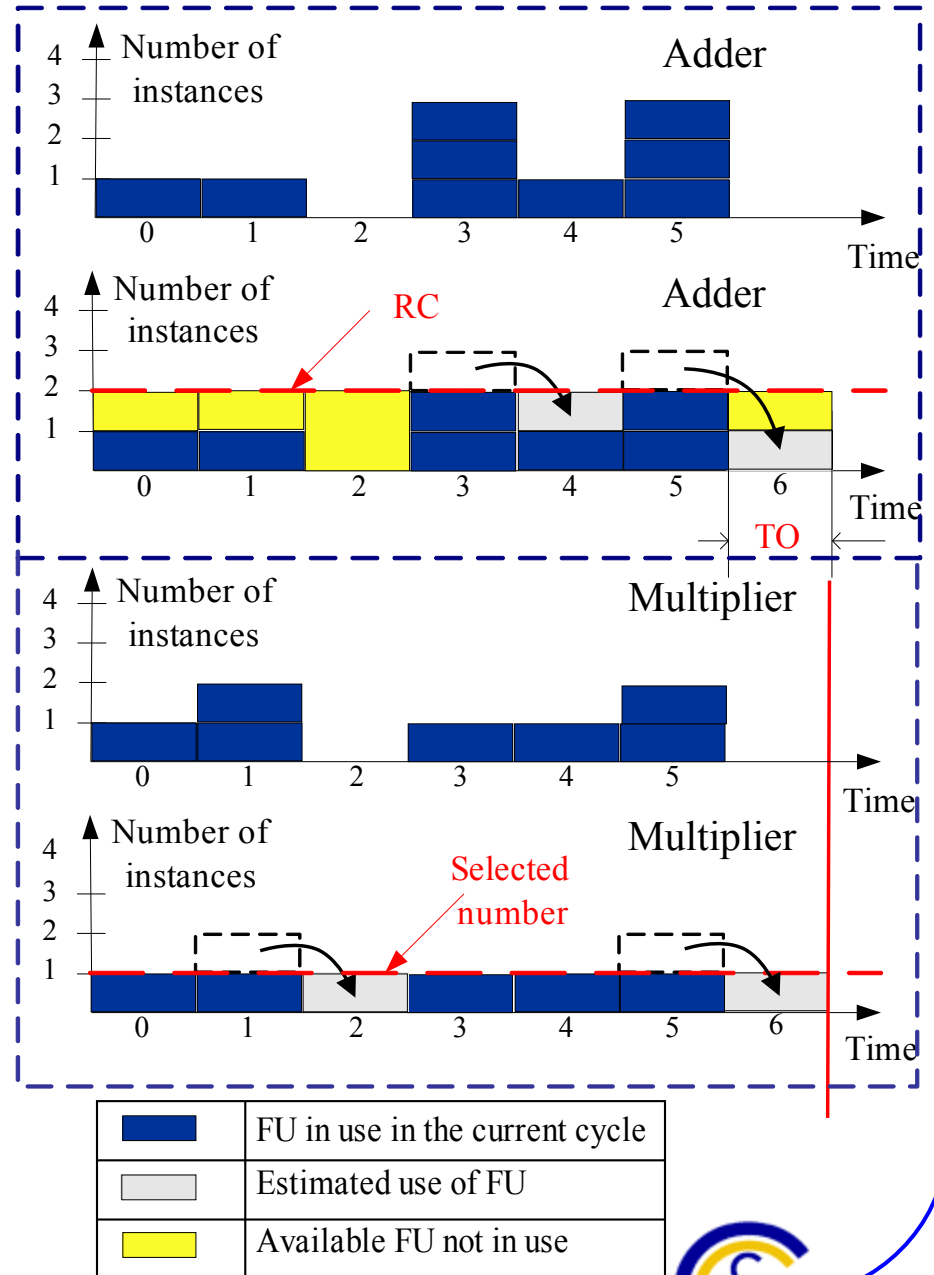
DP Creation and Evaluation



1. **Select basic blocks with max (frequency*length)**
2. **Derive component usage/cycle from schedule**
3. **Specify resource constraints (RC)**
4. **Derive timing constraints (TO) from RC**
5. **Recalculate number of components for TO**
6. **Repeat 5 for all unconstrained components**

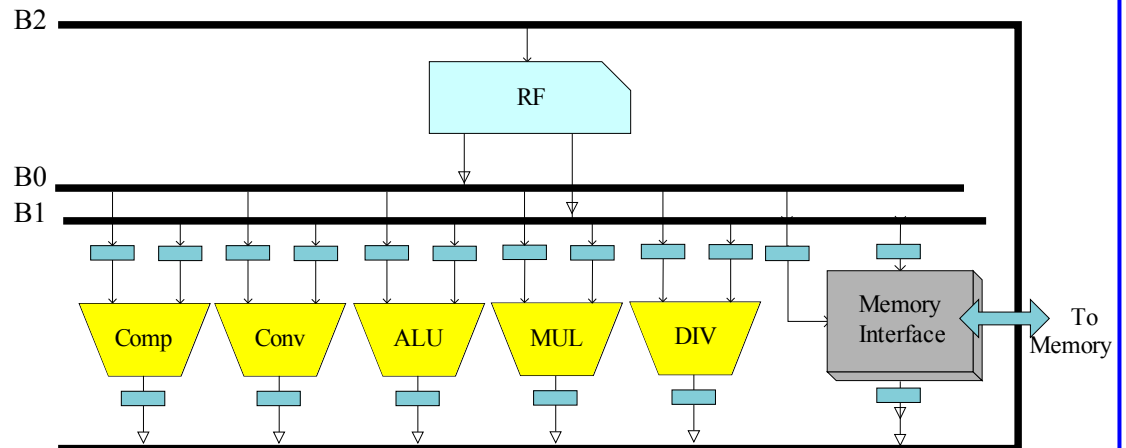
DP Generation Example

- **Initial DP consists of:**
 - 3 adders
 - 2 multipliers
- **Designer specified RC**
 - 2 adders
- **Estimation computes TO**
 - 1 cycle
- **Using computed TO decide number of multipliers**



Experimental Setup

- **Baseline: NISC-style general purpose processor**

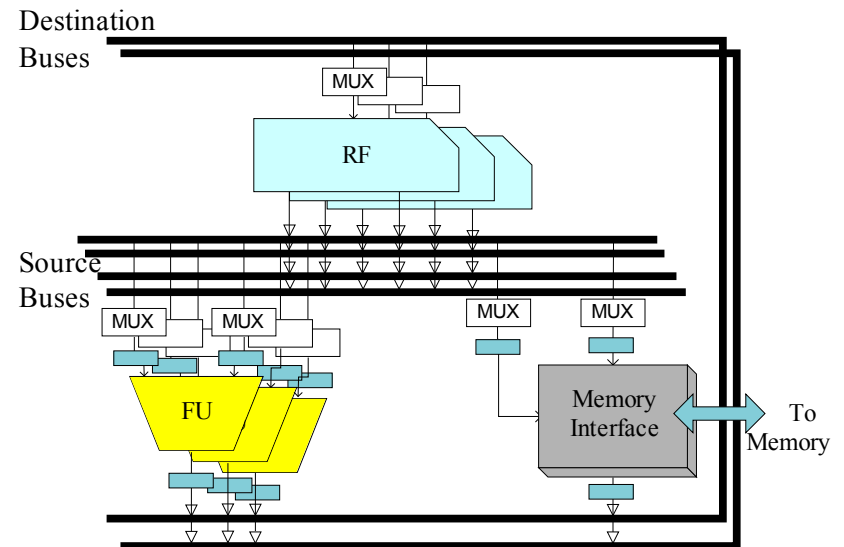


- **Generated DP**



- **Applications:**

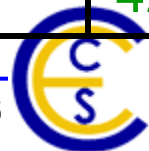
- bdist2 (MPEG2 encoder),
- dct32 (Mp3),
- Sort (bubble sort),
- Mp3 (fixed point)



Experimental Results

- **Speedup up to 21% (average 7%) compared to baseline**
- **DP generated in seconds, whereas manual design requires months**

Bench	LoC	Constraint RF	Pipe	Relative component reduction [%]			Bench. Speedup [%]	Algo. time [sec]
				FU	Bus	Pipe. Reg		
bdist2	61	3x1	N	40	-66.7	NA	14.2	0.2
		3x1	Y	40	-66.7	35.7	8.4	0.8
dct32	1006	3x1	N	20	-66.7	NA	-0.6	1.3
		3x1	Y	20	-66.7	14.3	21.4	2.3
Sort	33	2x1	N	40	0	NA	1.5	0.1
		2x1	Y	40	0	35.7	0.0	0.1
Mp3	13898	3x1	N	-20	-66.7	NA	7.8	15.6
		3x1	Y	-20	-66.7	-2.4	2.9	42.6



Conclusion

- **Automatic generation of datapath for a given C code**
 - Refined to satisfy given constraints
- **Runs in order of seconds**
- **Allows fast evaluation of each design with different constraints**
- **Future work**
 - Reduce area (complexity of FU, interconnect)
 - Add features (forwarding, special function units and memory hierarchy)



Thank You!

