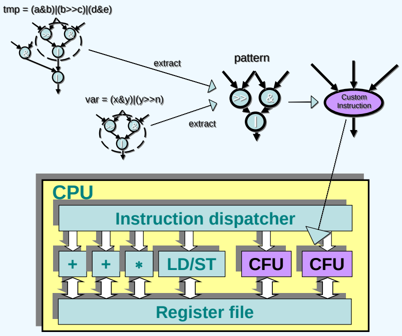
Satisfying Real-Time Constraints with Custom Instructions

Pan Yu Tulika Mitra

School of Computing, National University of Singapore

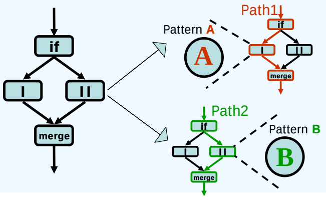
Custom Instructions

- Extend ISA with application-specific custom instructions
- Hardware implementation with custom functional units



Motivation: ACET *versus* WCET

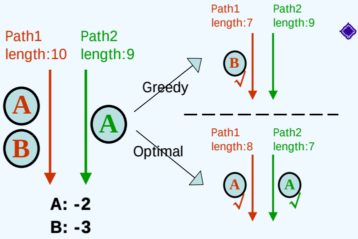
- Traditional custom instructions selection
  - Improves average-case execution time (ACET) thru profiling
- Custom instruction selection for real time tasks
  - Should improve **Worst-Case Execution Time (WCET)**



- ACET savings depend on execution frequency of paths
- WCET savings depend on which path is more critical (longer)

Challenges in improving WCET

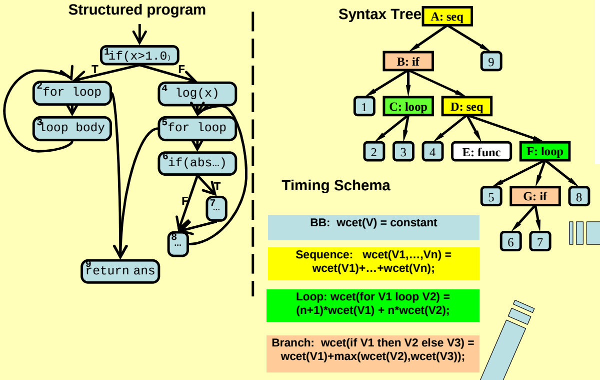
- Naïve approach: Greedily optimize current WCET path till it shifts to another path
  - Ignoring non-WCET paths may result in local optima



- Example of two patterns on different sides of a branch
  - **A** and **B** saves 2 and 3 cycles, respectively
  - We can select only one
  - Greedy: WCET = 9
Optimal: WCET = 8

WCET Estimation: Timing Schema

- Hierarchical syntax tree with basic blocks as leaf nodes and control structures as interior nodes



Timing Schema

BB: wcet(V) = constant

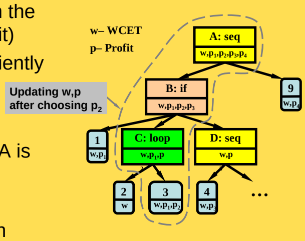
Sequence: wcet(V1,…,Vn) = wcet(V1)+…+wcet(Vn);

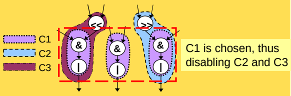
Loop: wcet(for V1 loop V2) = (n+1)*wcet(V1) + n*wcet(V2);

Branch: wcet(if V1 then V2 else V3) = wcet(V1)+max(wcet(V2),wcet(V3));

WCET Reduction with Heuristics

- Basic Heuristic:
  - Greedily select the pattern with the best WCET improvement (profit)
  - Update WCET and profits efficiently
- Improved heuristic:
  - When the current best pattern A is subsumed by other patterns
    - select A
    - select the subsuming pattern with maximum profit
  - Choose the best result from these searches





Optimal WCET Reduction: ILP Formulation

- Objective function: $\min : wcet_{main}$
- Structural Constraints:
  - Sequence ($V_1,\ldots,V_k$): $wcet_V = \sum_{i=1}^{k} wcet_{V_i}$
  - Branch if V1 then V2 else V3:

$$wcet_V \geq wcet_{V_1} + wcet_{V_2} \qquad wcet_V \geq wcet_{V_1} + wcet_{V_3}$$

  - Loop for V1 loop V2: $wcet_V = (n+1) \times wcet_{V_1} + n \times wcet_{V_2}$
  - Basic block: $wcet_V = T_V - (P_a \times s_{a.b} + \ldots + P_e \times s_{e.f})$
- Topological & Design Constraints:
  - At most one pattern covers an operation $s_{i_1.j_1} + \ldots + s_{i_k.j_k} \leq 1$
  - Area constraint:

$$S_i = 1 \quad \text{if} \sum_{j=1}^{n_i} s_{i.j} > 0 \qquad \sum_{i=1}^{N} (S_i \times R_i) \leq R$$
$$= 0 \quad \text{otherwise}$$

  - Number of custom instructions constraint: $\sum_{i=1}^{N} S_i \leq M$
- Variables:
  - S_i, s_{ij} – selection of ith pattern, selection of jth instance of ith pattern
  - F_{ij} – execution frequency of jth instance of ith pattern
  - P_i – performance gain by implementing ith pattern in hardware
  - R_i -- area requirement of ith pattern

Experimental Results

Program	Ptn.	Inst.	WCET Reduction		Time (s)	
			Heur	Opt.	Heur	Opt.
Adpcm	51	150	9%	9%	0.002	0.02
Blowfish	15	276	16%	16%	0.002	0.02
Compress	37	92	2%	2%	0.002	0.01
Crc	12	23	15%	15%	0.001	0.01
Djpeg	64	485	7%	7%	0.017	0.12
Gsmdec	158	2312	21%	22%	0.031	0.10
G721dec	73	180	4%	4%	0.006	0.03
Ndes	22	77	10%	10%	0.002	0.12
Rijndael	49	2520	16%	16%	0.034	1.25
Sha	9	40	12%	12%	0.001	0.01

Program	Ptn.	Inst.	WCET Reduction		Time (s)	
			Heur	Opt.	Heur	Opt.
Adpcm	101	258	14%	14%	0.005	0.04
Blowfish	56	1221	39%	39%	0.012	11.1
Compress	141	248	6%	6%	0.003	0.01
Crc	24	39	17%	17%	0.001	0.30
Djpeg	226	1056	11%	11%	0.028	0.28
Gsmdec	796	6782	26%	26%	0.064	0.05
G721dec	220	392	11%	11%	0.010	0.03
Ndes	77	182	17%	18%	0.003	0.03
Rijndael	156	9032	39%	39%	0.096	913
Sha	47	148	31%	31%	0.002	0.01

Significant WCET improvement by utilizing custom instructions

Heuristic is much faster than optimal solution

Heuristic achieves optimal WCET reduction in most cases

Heuristic is scalable with larger problem sizes

NUS National University of Singapore

School of Computing