Towards Multi-Objective Dynamic SPM Allocation

Shashank Jadhav Heiko Falk

shashank.jadhav / heiko.falk@tuhh.de

Institute of Embedded Systems, TUHH

21st International Workshop on Worst-Case Execution Time Analysis (WCET), Vienna, Austria

July 11, 2023

Overview

1. **Motivation**
2. Dynamic SPM Allocation (DSA)
3. Multi-Objective DSA-based Optimization
4. Evaluation
5. Conclusion

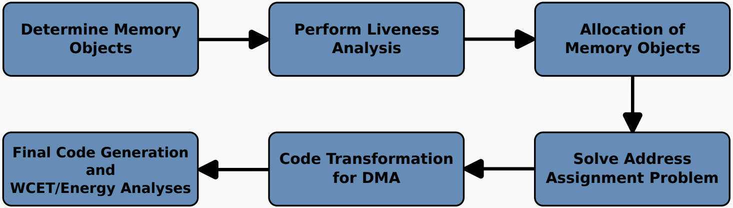
Motivation

- Worst-Case Execution Time (WCET)
- Energy consumption
- Static SPM allocation constrained by small SPM size

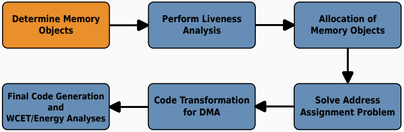
Overview

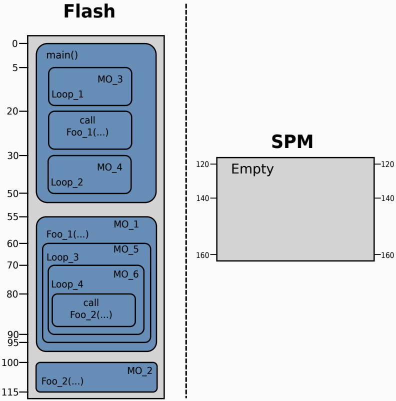
1. Motivation
2. **Dynamic SPM Allocation (DSA)**
3. Multi-Objective DSA-based Optimization
4. Evaluation
5. Conclusion

Dynamic SPM Allocation (DSA)



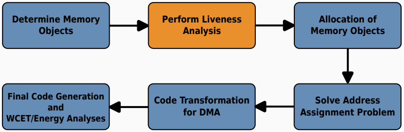
DSA: Memory Objects

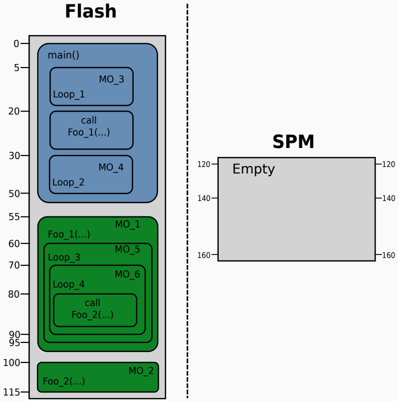




Memory Objects:
- -- *Functions*: Foo_1() and Foo_2()
- -- *Loops*: Loop_1, Loop_2, Loop_3, and Loop_4

DSA: Liveness Analysis

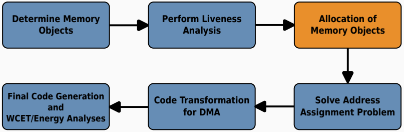


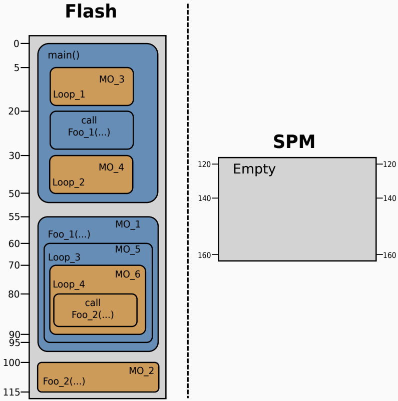


Memory Objects:
-- *Functions*: Foo_1() and Foo_2()
-- *Loops*: Loop_1, Loop_2, Loop_3, and Loop_4

Liveness conflicts:
-- Foo_2(), Loop_3, and Loop_4
-- Foo_1() and Foo_2()

DSA: Memory Objects Allocation



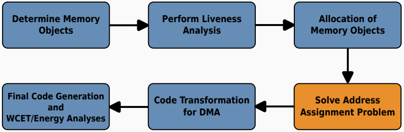


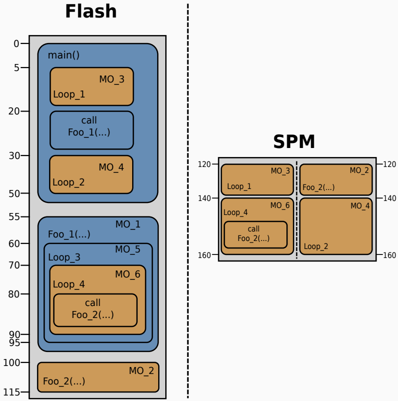
Memory Objects:
- -- *Functions*: Foo_1() and Foo_2()
- -- *Loops*: Loop_1, Loop_2, Loop_3, and Loop_4

Liveness conflicts:
- -- Foo_2(), Loop_3, and Loop_4
- -- Foo_1() and Foo_2()

Memory Object Allocation:
- -- *Flash*: Foo_1() and Loop_3()
- -- *SPM*: Foo 2(), Loop_1, Loop_2, and Loop_4

DSA: Address Assignment





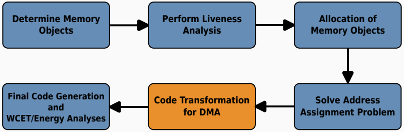
Memory Objects:
-- *Functions*: Foo_1() and Foo_2()
-- *Loops*: Loop_1, Loop_2, Loop_3, and Loop_4

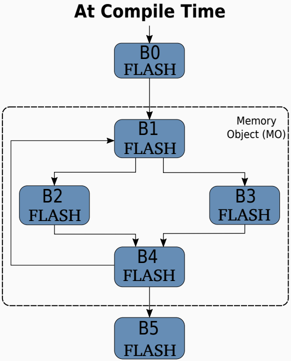
Liveness conflicts:
-- Foo_2(), Loop_3, and Loop_4
-- Foo_1() and Foo_2()

Memory Object Allocation:
-- *Flash*: Foo_1() and Loop_3()
-- *SPM*: Foo 2(), Loop_1, Loop_2, and Loop_4

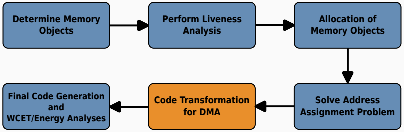
Address Assignment:
-- First-Fit Heuristic
-- Best-Fit Heuristic

DSA: Code Transformation

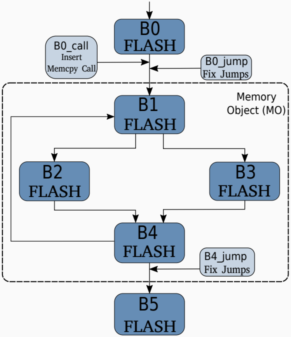




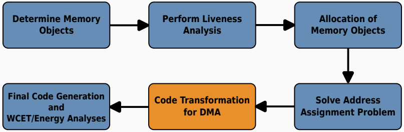
DSA: Code Transformation



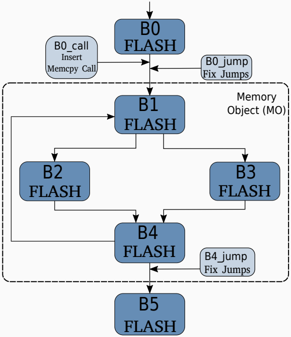
At Compile Time



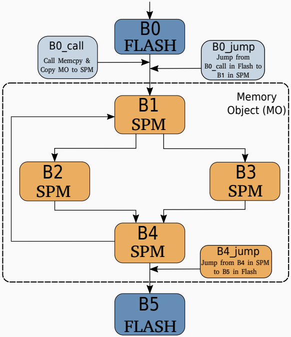
DSA: Code Transformation



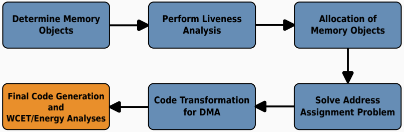
At Compile Time

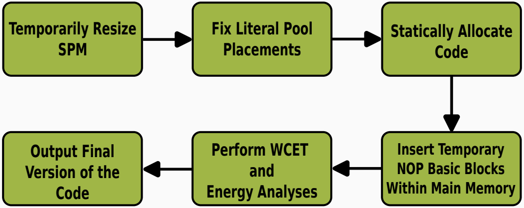


During Runtime



DSA: Code Generation and Analyses





Overview

Multi-Objective Optimization Problem

- **Search Space:**
  - $x \in X \in \{0,1\}^d$

Multi-Objective Optimization Problem

- **Search Space:**
  - $x \in X \in \{0, 1\}^d$
- **Objective Space:**
  - $\Theta = \{F(x) = (F_1(x), F_2(x)) | x \in X\}$
    Where, $F_1(x) =$ WCET objective and $F_2(x) =$ Energy objective

Multi-Objective Optimization Problem

- **Search Space:**
  - $x \in X \in \{0,1\}^d$
- **Objective Space:**
  - $\Theta = \{F(x) = (F_1(x), F_2(x)) | x \in X\}$
    Where, $F_1(x)$ = WCET objective and $F_2(x)$ = Energy objective
- **Minimization function:**
  - $\min_{x \in X} F(x)$

Multi-Objective Optimization Problem

- **Search Space:**
  - $x \in X \in \{0,1\}^d$
- **Objective Space:**
  - $\Theta = \{F(x) = (F_1(x), F_2(x)) | x \in X\}$
    Where, $F_1(x)$ = WCET objective and $F_2(x)$ = Energy objective
- **Minimization function:**
  - $\min_{x \in X} F(x)$
- **Search Space Constraint:**
  - $x_{(F+1):(F+L)} = x_{(F+1):(F+L)} + \tau$
    Where,

$$\tau_l = \begin{cases} 1, & \text{if } x_{F+l} = 0 \ \& \ (\exists f \mid \lambda_{F+l} \subseteq \lambda_f \in \Lambda_{1:F}) \ \& \ x_f = 1 \\ 0, & \text{otherwise} \end{cases}$$

Multi-Objective Optimization Problem

- **Search Space:**
  - $x \in X \in \{0,1\}^d$
- **Objective Space:**
  - $\Theta = \{F(x) = (F_1(x), F_2(x)) | x \in X\}$
    Where, $F_1(x)$ = WCET objective and $F_2(x)$ = Energy objective
- **Minimization function:**
  - $\min_{x \in X} F(x)$
- **Search Space Constraint:**
  - $x_{(F+1):(F+L)} = x_{(F+1):(F+L)} + \tau$
    Where,

$$\tau_l = \begin{cases} 1, & \text{if } x_{F+l} = 0 \text{ \& } (\exists f \mid \lambda_{F+l} \subseteq \lambda_f \in \Lambda_{1:F}) \text{ \& } x_f = 1 \\ 0, & \text{otherwise} \end{cases}$$

- **Address Assignment Algorithm Constraint:**
  - $(\mathcal{T} - \eta) = 0$

Overview

1 Motivation

2 Dynamic SPM Allocation (DSA)

3 **Multi-Objective DSA-based Optimization**

- Multi-Objective Optimization Problem

4 Evaluation

5 Conclusion

Metaheuristic Algorithms

To solve multi-objective DSA-based optimization problem, we use:

- Flower Pollination Algorithm (FPA)
- Strength Pareto Evolutionary Algorithm (SPEA)

Algorithm Multi-Objective DSA-based optimization

```
Collect memObj, perform Liveness Analysis, and randomly initialize initial population of size N
for n = 1 : N do
    Generate DSA code
while Stopping criteria is not reached do
    Update Individual using respective update operators
    for Each updated Individual do
        Generate DSA code
    Update to next generation using selection operator
return Pareto-optimal solution set
```

Comparison between...

- Proposed multi-objective DSA-based optimization (MO_D)
  −>Solved using:
  - FPA
  - SPEA
- Multi-objective static SPM allocation-based optimization (MO_S)
  −>Solved using:
  - FPA
  - SPEA
- ILP-based single objective dynamic SPM allocation (SO_D)

Overview

1. Motivation
2. Dynamic SPM Allocation (DSA)
3. Multi-Objective DSA-based Optimization
4. **Evaluation**
   - Pareto fronts
5. Conclusion

Pareto fronts

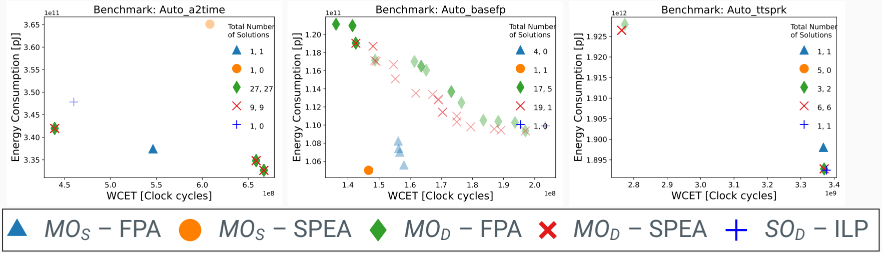


Figure 1: Solutions Obtained from MO_S, MO_D, and SO_D optimization runs

Pareto fronts

The following percent of solutions were on the final Pareto front

- MO_S–FPA: 3.62%
- MO_S–SPEA: 5.26%
- SO_D–ILP: 0.66%
- MO_D–FPA: 70.4%
- MO_D–SPEA: 20.1%

–> MO_D–FPA found most number of solution on the final Pareto front

Quality Indicators

- Coverage: $\mathcal{C} = 1 - \frac{|\{a \in A : \exists p \in \mathcal{P}, a \preceq p\}|}{|A|}$
- Non-Dominance Ratio: $NDR = \frac{|\mathcal{P} \cap A|}{|\mathcal{P}|}$
- Non-Dominated Solutions: $NDS = \frac{|a \in A : a \in \mathcal{P}|}{|A|}$

Quality Indicators

- Coverage: $\mathcal{C} = 1 - \frac{|\{a \in A : \exists p \in \mathcal{P}, a \preceq p\}|}{|A|}$
- Non-Dominance Ratio: $NDR = \frac{|\mathcal{P} \cap A|}{|\mathcal{P}|}$
- Non-Dominated Solutions: $NDS = \frac{|a \in A : a \in \mathcal{P}|}{|A|}$

From overall Evaluations, in terms of Quality Indicators:

- MO_D performed much better than SO_D
- MO_D performed slightly better than MO_S

- Coverage: $\mathcal{C} = 1 - \frac{|\{a \in A : \exists p \in \mathcal{P}, a \preceq p\}|}{|A|}$
- Non-Dominance Ratio: $NDR = \frac{|\mathcal{P} \cap A|}{|\mathcal{P}|}$
- Non-Dominated Solutions: $NDS = \frac{|a \in A : a \in \mathcal{P}|}{|A|}$

From overall Evaluations, in terms of Quality Indicators:

- MO_D performed much better than SO_D
- MO_D performed slightly better than MO_S

Overheads due to dynamic copying in MO_D optimization run:

- WCET overheads on average: 24.39%
- Energy overheads on average: 22.65%

Overview

1. Motivation
2. Dynamic SPM Allocation (DSA)
3. Multi-Objective DSA-based Optimization
4. Evaluation
5. **Conclusion**

Conclusion

- Proposed compiler-level DSA-based multi-objective optimization
- WCC performs WCET and energy analysis of DSA code
- MO_D is solved using FPA and SPEA
- MO_D outperforms SO_D
- MO_D performs slightly better than MO_S

Future Work

- Reducing the WCET and energy overheads by using DMA
- Reducing the compilation time needed by multi-objective DSA-based optimization

Thank You