Analisi stato dell'arte efficienza energetica sistemi embedded

Pietro Ghiglio

April 21, 2020

Osservazioni

- Ottimizzare il tempo di esecuzione porta ad un aumento della potenza richiesta ma riduce l'energia totale consumata.
- Molte ottimizzazioni sono applicate anche dal compilatore.
- Nessuna regola riguardo la trasmissione di informazioni e l'utilizzo di periferiche.

Regole

Reference: Javier Corral-García - Analysis of Energy Consumption and Optimization Techniques for Writing Energy-Efficient Code

- Unsigned int invece che multipli booleans.
- Accesso per righe alle matrici.
- Passare parametri per reference.
- Function inlining.
- Continue/Break invece che eccezioni.
- Inizializzare invece che assegnare.

Regole(2)

Reference: Brandolese, Fornaciari - The Impact of Source Code Transformations on Software Power and Energy Consumption.

- Loop distribution.
- Array declaration sorting.
- Array scope modification.
- Temporary array insertion.
- Scalarization of array elements.
- Subroutines reordering.
- Conditional expressions reordering.

Regole - Multi threads

Reference: Yunsi - Energy-Optimizing Source Code Transformations for Operating System-Driven Embedded Software

Estrarre informazioni riguardo all'IPC analizzando le chiamate a PThread. Utilizzo di un simulatore (EMSIM).

- Merge dei processi
- Vettorizzazione dei messaggi.
- Selezione del meccanismo di IPC (pipe, shared mem, messages).

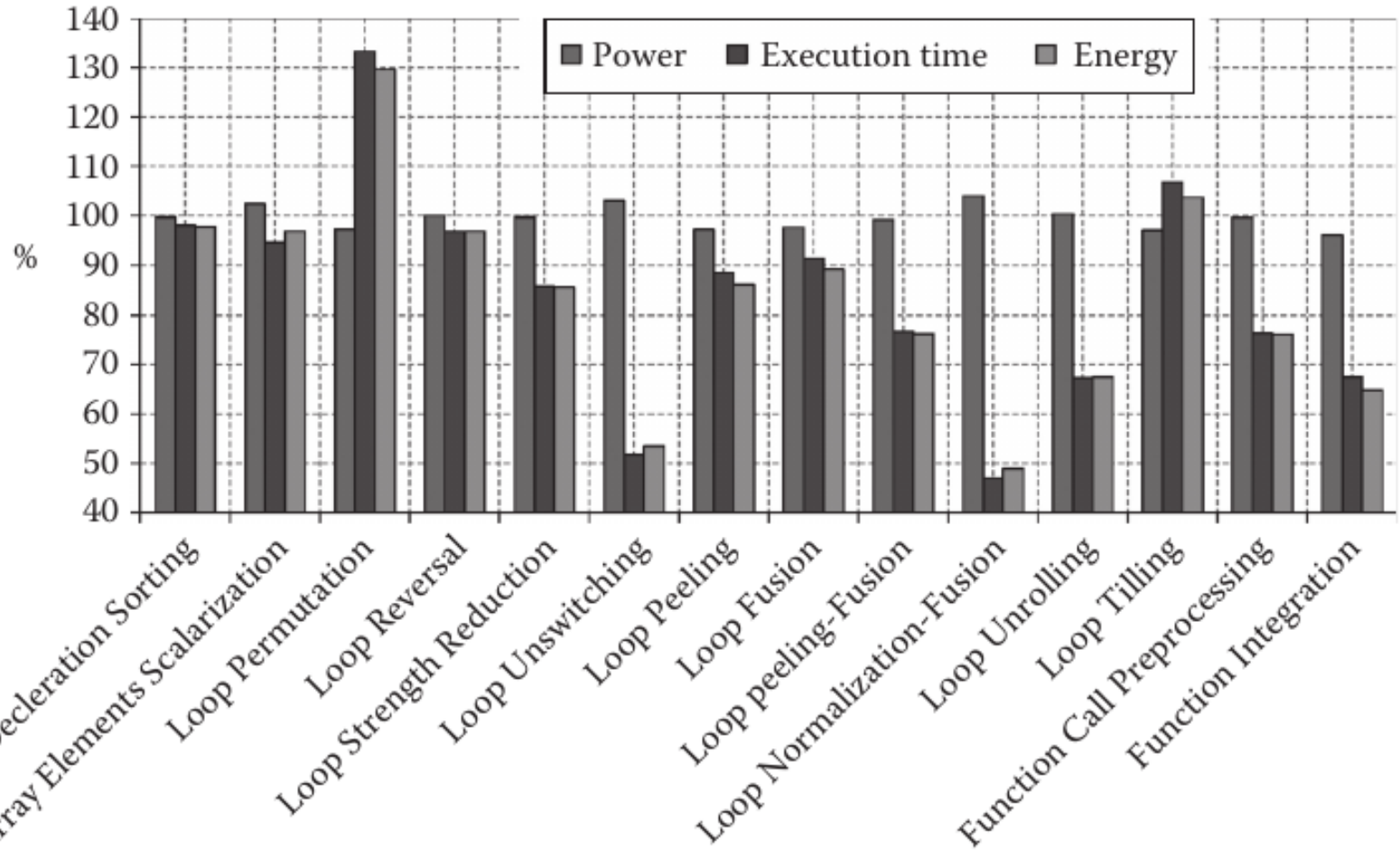
Regole - Architecture specific

Reference: link

- Far corrispondere tipi di dato e architettura (ex. 32 bit arch→variabili a 32 bit).
- Rendere dimensioni degli elementi di un array una potenza di due: semplifica il calcolo degli offset.
- Ordine degli argomenti passati a funzione.
- Direttive al compilatore (funzione pura, puntatore unico, promise)

Loop transformations

Reference: Mostafa - Embedded Systems Code Optimization and Power Consumption



Dubbi

- Target architecture.
- Periferiche.
- Regole già presenti.