

Systemy wbudowane

Wykład 11: Metody kosyntezy systemów wbudowanych

Uprozczone metody kosyntezy

Założenia:

- Jeden procesor o znanych parametrach
- Znane parametry akceleratora sprzętowego

■ Vulcan

- Początkowo zakłada się implementację sprzętową całego systemu
- Redukcja kosztu poprzez przenoszenie obliczeń do CPU

■ COSYMA

- Implementacja w formie programu
- Przenoszenie obliczeń do akceleratora sprzętowego

Vulcan

- Specyfikacja w HardwareC
- Architektura: CPU + ASIC
- Podział specyfikacji na wątki obliczeniowe o znanym czasie wykonania
- Wątki komunikują się poprzez kolejki komunikatów
- Oszacowania kosztu i wydajności dla każdego wątku
- Rozwiązanie początkowe: wszystkie wątki bez ograniczeń czasowych w CPU, pozostałe w sprzęcie
- Przenoszenie kolejnych wątków ze sprzętu do CPU:
 - Jeśli przeniesienie nie naruszy ograniczeń czasowych to zmiana jest akceptowana
 - Krok jest powtarzany dla następników przeniesionego wątku

1/2/2012

S.Deniziak:Systemy wbudowane

3

COSYMA

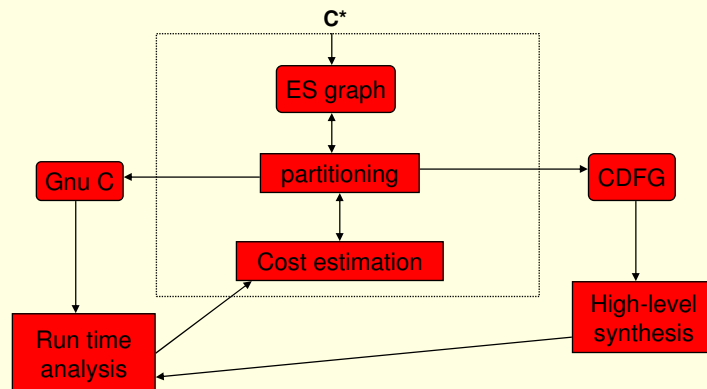
- Specyfikacja w C*
- Specyfikacja podzielona na bloki
- Bloki są przenoszone do sprzętu wg największego przyspieszenia
- Algorytm simulated annealing
- Komunikacja poprzez współdzieloną pamięć

1/2/2012

S.Deniziak:Systemy wbudowane

4

COSYMA design flow



1/2/2012

S.Deniziak:Systemy wbudowane

5

CoWare

- Specyfikacja w formie komunikujących się procesów
- Stopniowa rafinacja specyfikacji
- Co-synteza implementuje procesy
- Biblioteki zawierające charakterystyki procesorów, magistral, itp..

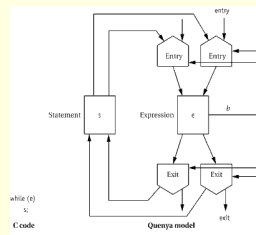
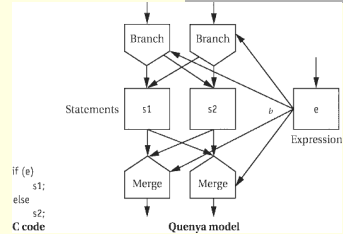
1/2/2012

S.Deniziak:Systemy wbudowane

6

LYCOS

- Jednorodny model dla specyfikacji w różnych językach: Quenya oparta o kolorowane sieci Petri
- Podział specyfikacji na bloki BSB
- Sekwencje BSB są prznoszone do sprzętu
- Poszukiwanie najszybszej kombinacji nie nakładających się w czasie sekwencji BSB nie przekraczającej danego kosztu



1/2/2012

S.Denziak:Systemy wbudowane

7

Metody kosyntezy dla systemów wieloprocesorowych

- Alokacja a szeregowanie:
 - Aby wykonać optymalną alokację należy mieć informacje uzyskane po uszeregowaniu
 - Aby wykonać uszeregowanie zadań należy alokować procesory
- Należy przyjąć pewne założenia aby problem dało się rozwiązać

1/2/2012

S.Denziak:Systemy wbudowane

8

Metoda ILP

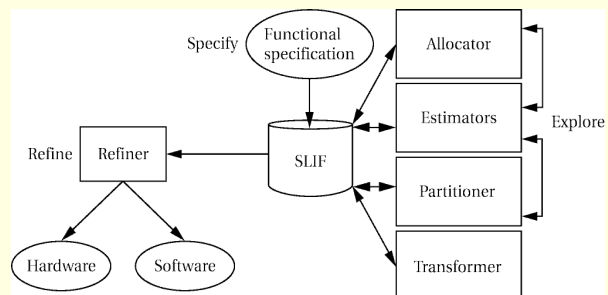
- Szeregowanie zadań w systemie wieloprocesorowym
- Zależności pomiędzy zadaniami, komunikacją i ograniczeniami są wyrażone jako zbiór równań i nierówności liniowych
- najlepsze rozwiązanie zostaje znalezione metodą programowania liniowego całkowitoliczbowego (ILP)
- Duża złożoność obliczeniowa

Metoda GCLP

- Specyfikacja: graf zadań
- Kryterium globalne:
 - Stopień w jakim implementacja zadania wpływa na wydajność całego systemu
- Kryterium lokalne:
 - Ocenia czy funkcje danego zadania lepiej zaimplementować w HW czy SW
- Synteza:
 - Obliczenie krytycznej ścieżki w systemie
 - Zmiana alokacji wybranego zadania na ścieżce w celu max. polepszenia parametrów systemu
 - Extremity: blokuje pewne nieopłacalne alokacje
 - Repeller: preferuje pewne alokacje
- 2 fazy:
 - Pętla wewnętrzna: szybkie i ogólne, przeszukiwanie przestrzeni rozwiązań
 - Pętla zewnętrzna: dokładne obliczenie parametrów rozwiązania (wykorzystywane w następnych wykonaniach pętli wewnętrznej)

SpecSyn

- Specyfikacja-eksploracja-rafinacja
- Specyfikacja w formie Statechart.
- Model SLIF



1/2/2012

S.Denziak:Systemy wbudowane

11

Rafinacja w SpecSyn

- Rafinacja sterowania:
 - Dodanie synchronizacji do funkcji rozdzielonych pomiędzy różne moduły
- Rafinacja danych:
 - Dodanie transmisji pomiędzy modułami operującymi na tych samych danych
- Rafinacja architektury:
 - Dodanie arbitrażu w przypadku konfliktu dostępu do wspólnych zasobów

1/2/2012

S.Denziak:Systemy wbudowane

12

Metoda rafinacyjna

- Architektura początkowa:
 - Każde zadanie wykonywane przez inny zasób (najszybciej wykonujący dane zadanie)
- Rafinacja:
 - Przenoszenie zadań wg kryterium najlepszego zysku
- Miara zysku:
 - Musi uwzględniać nie tylko lokalny ale i globalne polepszenie parametrów systemu

1/2/2012

S.Deniziak:Systemy wbudowane

13

COSYN

- Podział grafu zadań na klastry (redukcja dużych grafów zadań)
- Metoda konstrukcyjna: stopniowe budowanie architektury
- Uwzględnia wielokrotne wykonywanie zadań

```
assign priorities by deadline;
form association array;
form clusters of processes;
initialize architecture to empty;
allocate clusters;
foreach unallocated cluster Ci {
  form allocation array for Ci;
  foreach allocation in allocation array{
    schedule allocated clusters;
    evaluate completion time, energy, power.
    if deadline is met in best case then {
      save current architecture;
      break;
    }
    else save best allocation;
    tag cluster Ci as allocated;
  }
}
```

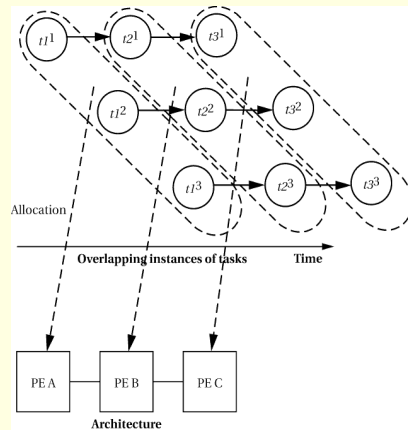
1/2/2012

S.Deniziak:Systemy wbudowane

14

Synteza przetwarzania potokowego

- Wymaga odpowiedniej alokacji (każde zadanie przyporządkowane do innego zasobu)
- Efektywne przy wielokrotnym wykonywaniu zadań



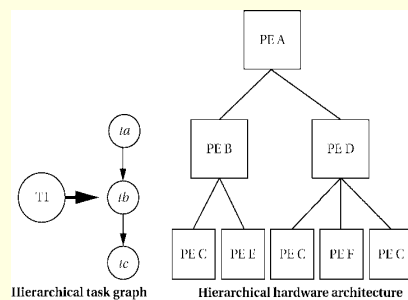
1/2/2012

S.Deniziak:Systemy wbudowane

15

Kosynteza hierarchiczna

- Węzłem w grafie zadań jest inny graf
- Moduł sprzętowy składa się z kilku mniejszych modułów.
- Kosynteza poprzez klasteryzację, alokację i szeregowanie



1/2/2012

S.Deniziak:Systemy wbudowane

16

Kosynteza wielokryterialna

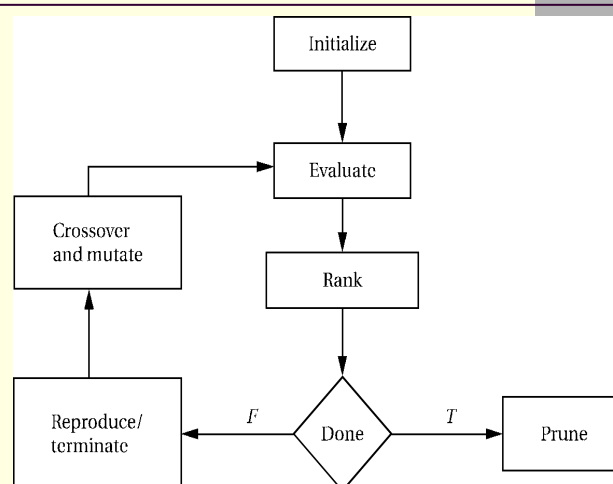
- Jednoczesna optymalizacja wg kilku kryteriów (koszt, szybkość, pobór mocy, itp.)
- Poprawa jednego parametru często pogarsza inne (optymalizacja Pareto)

1/2/2012

S.Deniziak:Systemy wbudowane

17

MOGAC – algorytm genetyczny



1/2/2012

S.Deniziak:Systemy wbudowane

18

Synteza systemów zorientowanych na przepływ sterowania

- Specyfikacja: CFSM
- Model sterowany zdarzeniami
- Ograniczenia: czas reakcji systemu
- Implementacja:
 - Sprzętowa: rejestry (aktualny stan) + logika kombinacyjna
 - Programowa: CFSM → s-graph → program C

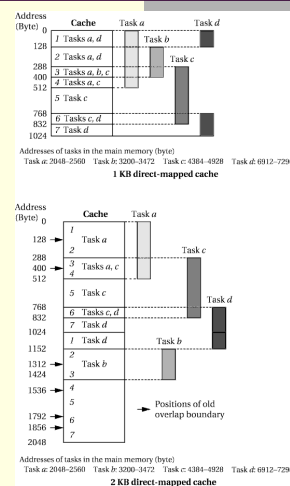
1/2/2012

S.Denziak:Systemy wbudowane

19

Synteza pamięci podręcznych

- Szybkość wykonywania zależy od efektywności pamięci podręcznej
- Rozmieszczenie procesów w pamięci powinno zależeć od organizacji cache
- Minimalizacja pamięci cache



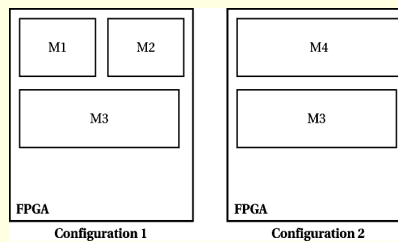
1/2/2012

S.Denziak:Systemy wbudowane

20

Kosynteza systemów rekonfigurowalnych

- FPGA – częściowo reprogramowalne
- Ten sam fragment sprzętu może implementować kilka funkcji
 - Należy uwzględnić rozmieszczenie modułów
- Szeregowanie musi uwzględnić czas reprogramowania



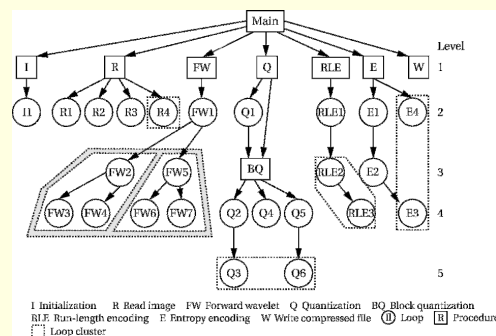
1/2/2012

S.Deniziak:Systemy wbudowane

21

Algorytmy syntezy systemów DR SOPC

- CORDS (wersja algorytmu MOGAC)
- Nimble:
 - Równoległość na poziomie instrukcji
 - Reprezentacja CFG
 - Przyspieszanie pętli



1/2/2012

S.Deniziak:Systemy wbudowane

22

Koniec