Systemy komputerowe

Lista zadań nr 3

Na ćwiczenia 19 marca 2025

Każde zadanie warte jest 1 punkt.

Zadanie 1. Przedstaw zasadę działania układu dzielącego dwie liczby bez znaku w wersji *nonrestoring division.* Uzasadnij poprawność dzielenia tą metodą. Wykonaj nietrywialny przykład dzielenia.

Wskazówka: "Appendix J", str. 4. — 6.

Zadanie 2. Przedstaw algorytm dzielenia metodą SRT (ang. *SRT division*). Uzasadnij jego poprawność oraz zademonstruj działanie na wybranym przez siebie przykładzie. Jaka motywacja stoi za wprowadzeniem tego algorytmu?

Wskazówka: "Appendix J", str. 45. — 47.

Zadanie 3. Rozważmy algorytm mnożenia tablicowego liczb 5-bitowych. Realizujący go układ cyfrowy (Appendix J. Fig. J.27) składa się z trzech sumatorów CSA oraz jednego sumatora RCA. Jest to układ kombinacyjny, tzn. wartości na jego wyjściach zależą jedynie od wartości podanych na wejściach; nie posiada on pamięci oraz nie jest synchronizowany sygnałem zegarowym.

1. Zaproponuj układ sekwencyjny, wykorzystujący pamięć oraz pracujący w kilku etapach (cyklach) wyznaczonych przez sygnał zegarowy, który mnoży dwie liczby 5-bitowe, ale używa tylko jednego sumatora CSA (i jednego RCA).
2. Układ z punktu 1. potrzebuje wielu cykli do wyliczenia wyników jednego mnożenia. Zaproponuj układ sekwencyjny, który ma tyle samo sumatorów CSA i RCA co oryginalny układ mnożenia tablicowego, ale w każdym cyklu (z wyjątkiem kilku początkowych) będzie wyprowadzał na wyjście wyniki mnożeń kolejnych par liczb podawanych na wejściu.

Uwaga: W tym zadaniu należy podać ideę układu i jego schemat wysokiego poziomu (w stylu Fig. J.27a). Nie wymagam schematu układu z dokładnością do bramki.

Zadanie 4. Przedstaw algorytm mnożenia oparty na systemie czwórkowym z wykorzystaniem kodowania Bootha (ang. *radix-4 Booth recording*). Uzasadnij jego poprawność oraz zademonstruj działanie na wybranym przez siebie przykładzie. Jaka motywacja stoi za wprowadzeniem tego algorytmu?

Wskazówka: "Appendix J", str. 48. — 49.

Zadanie 5. Dla poniższego programu narysuj graf przepływu sterowania. Następnie, dla każdej instrukcji l, podaj zbiory definicji osiągających tę instrukcję (ang. *Reaching Definition sets*), a dokładniej zbiory faktów prawdziwych na wejściu do instrukcji l (oznaczamy go $RD_{\circ}(l)$) oraz prawdziwych na wyjściu z tej instrukcji ($RD_{\bullet}(l)$).

$[x := 0]^1$
$[y := 1]^2$
$[i := 1]^3$
$while\ [i < z]^4\ do$
    $[t := x + y]^5$
    $[x := y]^6$
    $[y := t]^7$
    $[i := i + 1]^8$
od
$[y := x]^9$

Wskazówka: Zbiory w tym zadaniu wylicz ręcznie, tak jak na slajdach 19–24 z pliku slides1.pdf.

Zadanie 6. Kompilator wygenerował kod pośredni, w którym występuje następujący fragment. W jaki sposób zoptymalizować go pod względem wydajności? Wymyśl odpowiednią analizę przepływu danych: a) zdefiniuj ją słownie, b) zastanów się, w jaki sposób powinien wyglądać pojedynczy fakt w takiej analizie, c) napisz zbiory faktów prawdziwych na wejściu-do oraz wyjściu-z każdej instrukcji tego programu.

$$[x := 10]^1; [y := x + 10]^2; [z := y + x]^3$$