4.1. Bucle de control simple

4.1.1 Descrieri care utilizează construcții condiționale şi bucle

Pe lângă blocurile de instrucțiuni secvențiale, descrierile funcționale conțin în general atât construcții condiționale, cât şi construcții de buclare. Astfel, un algoritm de planificare realist trebuie să țină cont de aceste construcții.

Construcții condiționale

O construcție condițională este similară cu o instrucțiune *if* sau *case* a unui limbaj de programare. Aceasta are ca rezultat o serie de ramuri care sunt mutual exclusive. Figura 4. 1 prezintă un fragment al unei descrieri funcționale care conține o instrucțiune condițională *if*.

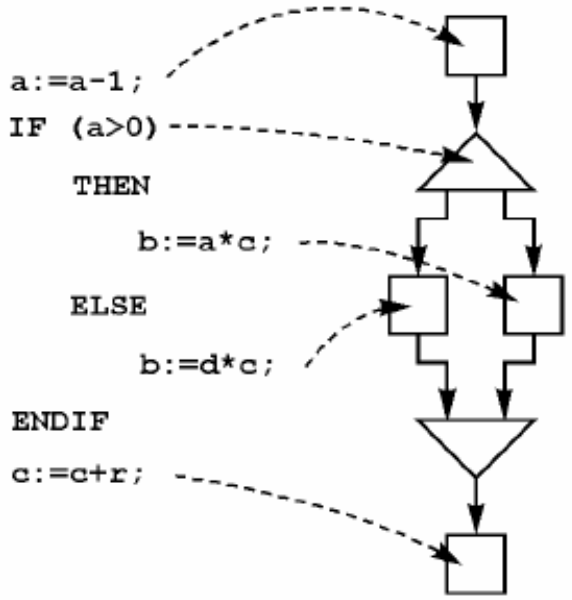

Figura 4. 1. GFCD pentru o descriere funcțională cu o construcție condițională if

Această descriere nu poate fi reprezentată numai printr-un graf al fluxului de date, fiind necesar un graf cu dependențe de control şi de date (de exemplu, GFCD). În acest exemplu, fluxul de control determină execuția celor patru grafuri ale fluxului de date reprezentate prin pătrate. Un algoritm eficient de planificare partajează resursele între operațiile mutual exclusive. De exemplu, va fi executată numai una din cele două operații de înmulțire din Figura 4. 1 pe durata fiecărei execuții a modelului. Astfel, algoritmul poate realiza planificarea ambelor operații de înmulțire în acelaşi pas de control, chiar dacă este disponibil un singur circuit de înmulțire în fiecare pas.

4.1.2 Construcții de buclare

O descriere funcțională poate conține construcții de buclare. De exemplu, într-un filtru pentru aplicații DSP, un set de acțiuni este executat în mod repetat pentru fiecare eşantion al şirului datelor de intrare. Această acțiune repetată este modelată printr-o construcție de buclare. În aceste descrieri, optimizarea corpului buclei poate îmbunătăți performanțele modelului. Construcțiile de buclare prezintă un paralelism potențial între diferitele iterații ale buclei. Ca urmare, pot fi executate mai multe iterații ale aceleiaşi bucle în mod concurent.

Planificarea unui corp al buclei diferă de planificarea unui graf al fluxului de date, prin faptul că trebuie să se considere paralelismul potențial între diferitele iterații ale buclei. Se utilizează Figura 4. 2 pentru a ilustra trei metode diferite de planificare a unei bucle.

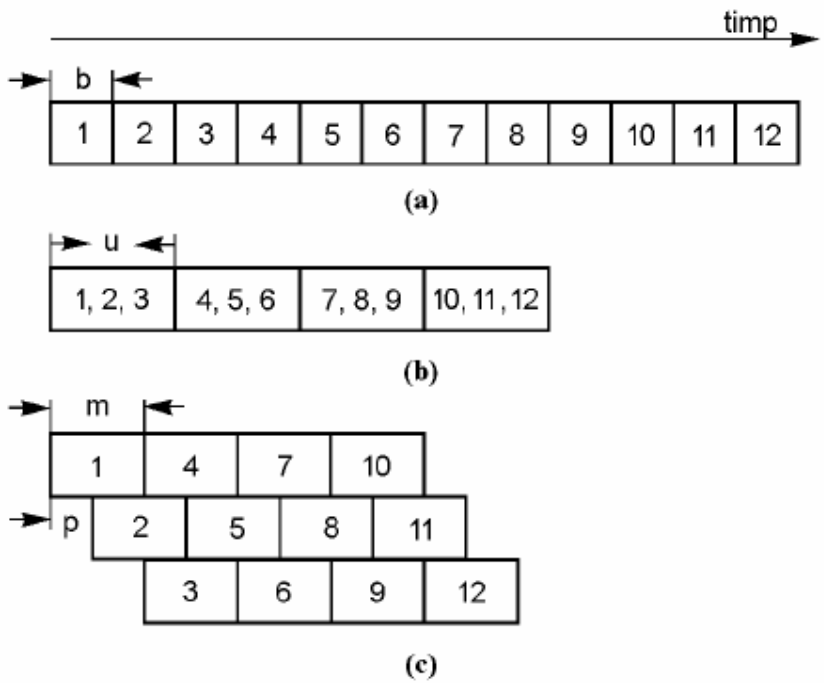

Figura 4. 2 Planificarea buclelor: (a) execuție secvențială; (b) expandarea parțială a buclei; (c) suprapunerea iterațiilor succesive ale buclei

Se presupune că bucla este finită şi constă din n iterații (n = 12 în Figura 4. 2). Prima şi cea mai simplă metodă presupune *execuția secvențială* a buclei şi planifică fiecare iterație a buclei în b paşi de control. Dacă cele 12 iterații sunt executate secvențial, timpul total de execuție este de $12b$ paşi de control, după cum se indică în Figura 4. 2(a). Celelalte două metode țin cont de paralelismul între iterațiile buclei.

A doua tehnică este numită *expandarea buclelor*, deoarece un anumit număr de iterații ale unei bucle sunt desfăşurate. Această acțiune are ca rezultat o buclă cu un corp al buclei de dimensiuni mai mari, dar cu un număr mai redus de iterații. Corpul buclei de dimensiuni mai mari permite o mai mare flexibilitate pentru compactarea planificării. În Figura 4. 2(b) sunt desfăşurate trei iterații într-o super-iterație, rezultând patru super-iterații. Fiecare super-iterație este planificată pe durata a u paşi de control. Astfel, dacă $u < 3b$, timpul total de execuție este mai mic decât $12b$ paşi de control. Translatarea limbajelor de descriere a unităților hardware 59

A treia metodă ține cont de paralelismul din interiorul buclelor, realizând *suprapunerea de tip pipeline a unor iterații succesive* ale unei bucle. Figura 4. 2(c) prezintă o buclă cu un corp care necesită m paşi de control pentru execuție. În cadrul suprapunerii iterațiilor, se inițiază o nouă iterație după fiecare p paşi de control, unde $p < m$, deci se suprapun iterații succesive. Timpul total de execuție este de $m + (n - 1) \times p$ paşi de control (Figura 4. 2). Expandarea buclelor este aplicabilă numai dacă contorul buclei este cunoscut în avans, dar suprapunerea iterațiilor poate fi aplicată atât pentru bucle cu un număr fix de iterații, cât şi pentru cele cu un număr de iterații variabil. Pentru ilustrarea acestor metode de execuție a buclelor, se alege un exemplu de graf al fluxului de date care conține iterații multiple. Figura 4. 3(a) prezintă graful corpului unei bucle care conține 17 operații identice, fiecare necesitând un pas de control pentru execuție. Arcele cu linii întrerupte indică dependențe ale datelor peste limitele buclei.
De exemplu, arcul de la nodul P la nodul J indică faptul că rezultatul generat de operația P în timpul unei iterații este utilizat de operația J în următoarea iterație. Se va realiza planificarea acestui graf al fluxului de date utilizând cele trei tehnici de planificare a buclelor.
În scopul evaluării calității planificărilor rezultate, se utilizează două măsuri de performanță: rata de utilizare a unităților funcționale, deci procentul de stări în care unitățile funcționale sunt utilizate pentru a executa o anumită operație, şi costul părții de control, care este măsurat prin numărul cuvintelor de control unice din unitatea de control.

În Figura 4. 3(b) se prezintă o planificare cu 6 paşi de control utilizând 3 unități funcționale. Deoarece o cale de date cu 3 unități funcționale necesită cel puțin 6 paşi de control pentru a executa 17 operații, iar lungimea căii critice a grafului este de 6, planificarea este optimă. Rata

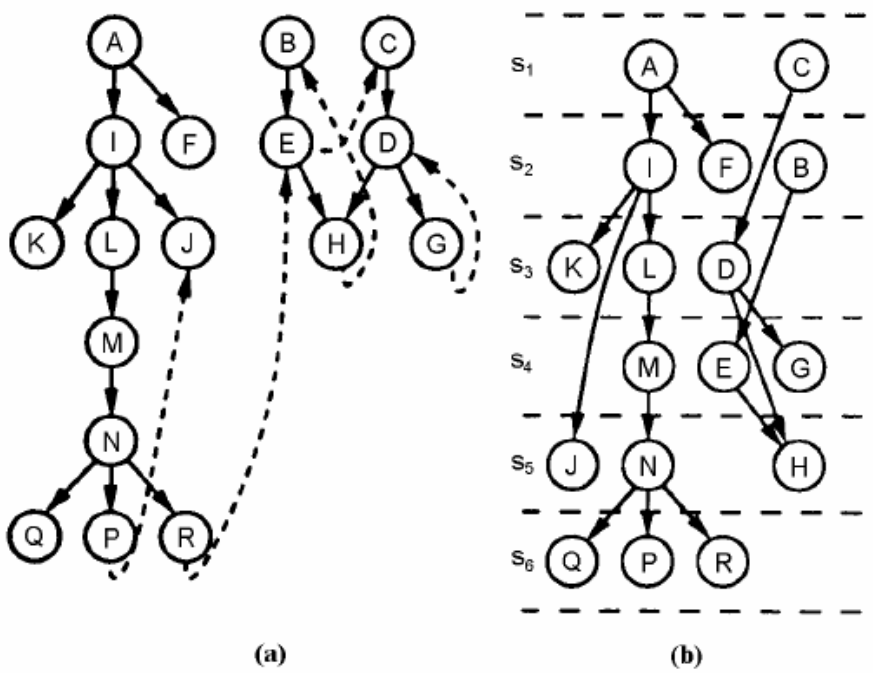

Figura 4. 3. Planificarea standard a unei bucle: (a) GFD cu dependențe între iterații; (b) planificare secvențială utilizând trei unități funcționale

de utilizare a unităţilor funcţionale este 17 / (3 × 6) = 17 / 18.
Translatarea limbajelor de descriere a unităţilor hardware 60 Costul părţii de control este de 6 cuvinte, presupunând că este necesar un cuvânt în unitatea de control pentru fiecare pas de control planificat. Performanţele obţinute pot fi îmbunătăţite numai dacă se ţine cont de paralelismul existent.

Translatarea limbajelor de descriere a unităţilor hardware 61 În contextul sintezei de nivel înalt, trebuie să se considere modificarea costului părţii de control atunci când se efectuează aceste optimizări. Atât expandarea buclelor, cât şi suprapunerea ciclurilor determină creşterea costului părţii de control. Trebuie efectuat un compromis între viteza de execuţie şi costul părţii de control.

Planificarea unei bucle expandate este relativ simplă, deoarece algoritmii de planificare descrişi pot fi aplicaţi grafului expandat al fluxului de date. Singura problemă constă în gradul de expandare al buclei. Performanţele unor algoritmi cu o complexitate ridicată a calculelor (de exemplu, metoda programării liniare) vor fi reduse în cazul unui grad de expandare ridicat al buclei. Pe de altă parte, planificarea unei bucle cu suprapunerea ciclurilor este mai complexă.

Pentru un număr fixat al paşilor de control între iteraţiile succesive, k, se poate extinde algoritmul de planificare bazat pe liste pentru a planifica operaţii cu restricţii de resurse. Într-un pas de control q, astfel încât $q > k$, operaţiile din diferitele iteraţii ale buclei sunt executate concurent. În consecinţă, trebuie prevăzute resursele necesare pentru execuţia diferitelor operaţii în toate iteraţiile concurente

Bibliografie:

1. Istvan Sztojanov, Sever Paşca, Elisabeta Buzoianu, Aplicaţii hardware şi software cu microcontrolerul PIC12F675, Editura Cavallioti, ISBN 978-973-7622-54-9, Bucuresti 2008
2. Istvan Sztojanov, Alexandru Vasile, Elisabeta Buzoianu, Sever Paşca, *Programarea microcontrolerelor din familia Intel, Aplicaţii practice hardware cu 80C552*, Editura Man-Dely, ISBN 973-85681-5-3, Bucureşti 2004.
3. http://vega.unitbv.ro/~romanca/EmbSys/
4. http://facultate.regielive.ro/cursuri/electronica/
5. www.microcip.com
6. Andrei Drumea, Teza de doctorat, UPB 2009
7. BARUCH Zoltan, Translatarea limbajelor de descriere a unităţilor hardware, Referat de doctorat Cluj Napoca 2008