

## G.1. TABLICE REZERWACJI PRZETWARZANIA POTOKOWEGO

Głównym problemem w projektowaniu przetwarzania potokowego jest zapewnienie najwyższej przepustowości zgodnej z unikaniem zagrożeń strukturalnych. Oznacza to, że chcielibyśmy przesyłać rozkazy do potoku z najwyższą szybkością, jaką można osiągnąć bez kolizji wynikających ze współdzielenia zasobu. Jedną z najstarszych technik zastosowanych w tym celu jest tablica rezerwacji ([DAVI71], [DAVI75], [PATE76]).

Tablica rezerwacji to diagram czasowy, który określa przepływ danych przez potok oraz jakie zasoby są wymagane w każdym przedziale czasu przez rozkaz przepływający przez potok.

### Tablice rezerwacji w dynamicznym przetwarzaniu potokowym

Najpierw przeanalizujemy bardziej ogólny przypadek dynamicznego przetwarzania potokowego. Dynamiczne przetwarzanie potokowe to takie, które można ponownie skonfigurować w celu obsługi różnych funkcjonalności w różnym czasie. Ponadto taki potok może obejmować zarówno sprzężenie do przodu, jak i sprzężenie zwrotne. Większość potoków rozkazów nie będzie charakteryzować się aż taką elastycznością, aczkolwiek niektóre implementacje mogą zawierać pewne cechy przetwarzania dynamicznego. W każdym razie statyczne przetwarzanie potokowe, wykonujące stałą funkcję bez sprzężenia zwrotnego lub sprzężenia do przodu, jest tylko szczególnym przypadkiem, w którym obowiązują te same zasady.

Na rysunku G.1a przedstawiono schemat wielofunkcyjnego dynamicznego przetwarzania potokowego. Dwie tablice rezerwacji zostały pokazane na rysunkach G.1b oraz G.1c i odpowiadają funkcji  $X$  oraz funkcji  $Y$ . W tablicy rezerwacji wiersze odpowiadają zasobom (w tym przypadku etapom przetwarzania potokowego), kolumny jednostkom czasu, natomiast wpis w  $i$ -tym wierszu i  $j$ -ej kolumnie oznacza, że  $i$ -ta *stacja* jest zajęta w  $j$ -ym czasie.

Liczba jednostek czasu (cykli zegarowych) między dwoma zdarzeniami zainicjowania potoku to **opóźnienie** między nimi. Każde dwie lub więcej prób zainicjowania użycia tego samego zasobu potoku w tym samym czasie spowoduje **kolizję**. Łatwo zauważyć, że kolizja wystąpi, gdy dwa rozkazy zostaną zainicjowane z opóźnieniem równym odległości między dwoma wpisami w danym wierszu. Podczas analizy tablicy rezerwacji można określić listę takich zabronionych opóźnień oraz zbudować **wektor kolizji**.<sup>1</sup>

$$(C_1, C_2, \dots, C_n)$$

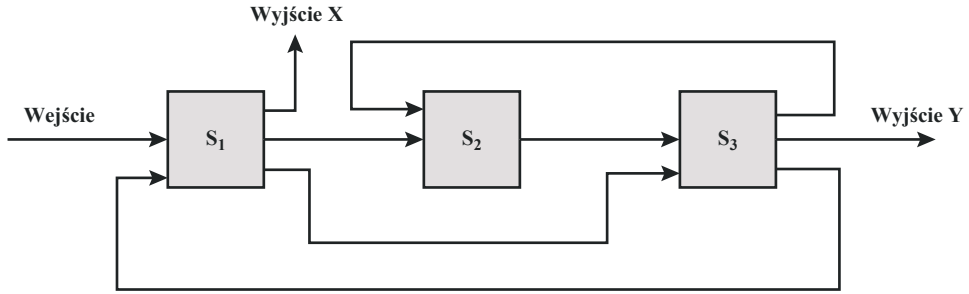
gdzie:

$C_i = 1$ , jeśli  $i$  jest opóźnieniem zabronionym, czyli zainicjowanie rozkazu potoku  $i$  jednostek czasu po poprzednim rozkazie powoduje kolizję zasobów.

$C_i = 0$ , jeśli  $i$  jest opóźnieniem dozwolonym.

$n$  = największa wartość na liście kolizji.

<sup>1</sup> W symulacji zamieszczonej w zasobach online, do której odwołuje się ta książka, wektor kolizji został zdefiniowany jako zaczynający się od  $C_0$ . Jednakże opóźnienie równe 0 zawsze spowoduje kolizję, zatem ta konwencja nie jest stosowana w literaturze. Ponadto niektóre techniki przetwarzania tablic rezerwacji wykorzystują odwrotną kolejność wektora kolizji:  $(C_1, C_2, \dots, C_n)$ .



(a) Potok trzyetapowy

	czas →							
	1	2	3	4	5	6	7	8
S <sub>1</sub>	X					X		X
S <sub>2</sub>		X		X				
S <sub>3</sub>			X		X		X	

(b) Tablica rezerwacji dla funkcji X

	czas →					
	1	2	3	4	5	6
S <sub>1</sub>	Y				Y	
S <sub>2</sub>			Y			
S <sub>3</sub>		Y		Y		Y

(c) Tablica rezerwacji dla funkcji Y

**Rysunek G.1.** Przetwarzanie potokowe ze sprzężeniem zwrotnym i sprzężeniem do przodu dla dwóch różnych funkcji

Pomimo że nie jest to widoczne w tych przykładach, to może istnieć wiele kolejnych wpisów w wierszu. Odpowiada to przypadkowi, w którym dany etap przetwarzania potokowego wymaga wielu jednostek czasu. Analogicznie także w kolumnie może istnieć wiele wpisów, co wskazuje na równoległe użycie wielu zasobów.

W celu określenia, czy dane opóźnienie jest zabronione lub dozwolone możliwe jest użycie dwóch kopii wzorca tablicy rezerwacji, dostawienie jednej z nich po prawej stronie i sprawdzenie, czy przy danym opóźnieniu występują kolizje. Jak widać na rysunku G.2, opóźnienia 2 i 5 są zabronione. Na rysunku oznaczenie  $X_1$  odnosi się do pierwszego zainicjowania funkcji X, natomiast  $X_2$  do drugiego zainicjowania.