Analizując dokładniej, możemy również zauważyć, że opóźnienia 2, 4, 5 i 7 są również zabronione.

Dozwolonymi opóźnieniami są 1, 3 i 6 oraz oczywiście każde opóźnienie większe niż 7. Zatem wektor kolizji dla funkcji X to:

$$C_X = (0101101)$$

	1	2	3	4	5	6	7	8	9	10
S_1	X_1		X_2			X_1		X_1, X_2		X_2
S_2		X_1		X_1, X_2		X_2				
S_3			X_1		X_1, X_2		X_1, X_2		X_2	

(a) Kolizja z opóźnieniem 2

	1	2	3	4	5	6	7	8	9	10	11
S_1	X_1					X_1, X_2		X_1			
S_2		X_1		X_1			X_2		X_2		• • •
S_3			X_1		X_1		X_1	X_2		X_2	

(b) Kolizja z opóźnieniem 5

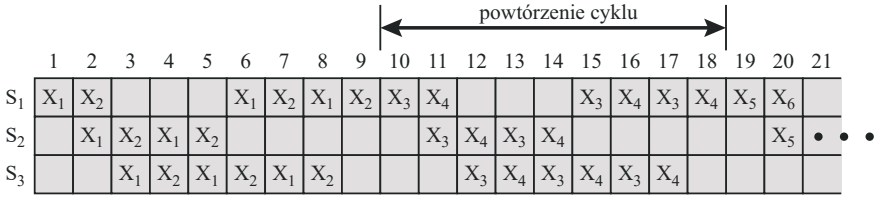
Rysunek G.2. Kolizje z zabronionymi opóźnieniami 2 i 5 dla potoku z rysunku G.1 i dla funkcji X

Interpretację wektora kolizji należy przeprowadzać w ostrożny sposób. Pomimo że opóźnienie równe 1 jest dozwolone, to nie można wydać sekwencji rozkazów, z których każdy charakteryzuje się opóźnieniem wynoszącym 1 względem poprzedniego rozkazu. By to zobrazować, załóżmy, że trzeci rozkaz miałaby opóźnienie wynoszące 2 w stosunku do rozkazu pierwszego, co jest zabronione. Zamiast tego potrzebujemy sekwencji opóźnień, które są dopuszczalne w odniesieniu do wszystkich poprzednich rozkazów. **Cykl opóźnienia** to sekwencja opóźnień, która powtarza tę samą podsekwencję w nieskończoność.

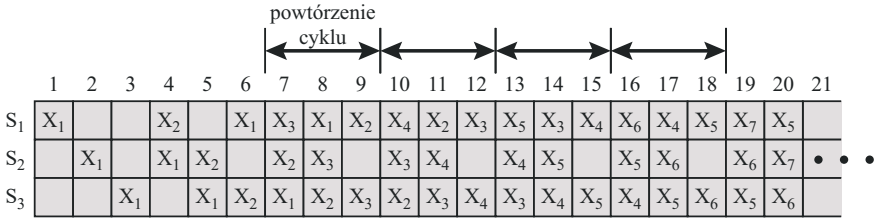
Rysunek G.3 ilustruje cykle opóźnień dla analizowanego przykładu, które nie powodują kolizji. Z rysunku G.3a wynika na przykład, że inicjowanie kolejnych nowych zadań jest oddzielone naprzemiennie jednym i ośmioma cyklami.

Przykład przetwarzania potokowego rozkazu

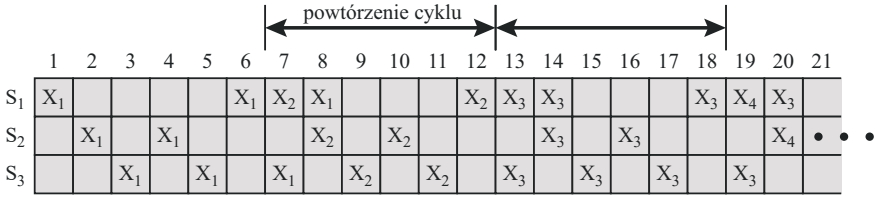
Przeanalizujmy teraz budowę tablicy rezerwacji dla typowego potoku. Rozważmy następujący przykład. Na rysunku G.4 przedstawiono potok wraz z przybliżoną ilością czasu wymaganą do zrealizowania każdego etapu. Rezultat prostego opracowania tablicy rezerwacji pokazano na rysunku G.5a. W celu opracowania takiej tablicy używamy ustalonej częstotliwości zegara i synchronizujemy całe przetwarzanie z tym



(a) Cykl opóźnienia (1, 8) = 1, 8, 1, 8, ... ze średnim opóźnieniem 4,5



(b) Cykl opóźnienia (3) = 3, 3, 3, 3, ... ze średnim opóźnieniem 3



(c) Cykl opóźnienia (6) = 6, 6, 6, 6, ... ze średnim opóźnieniem 6

Rysunek G.3. Dopuszczalne opóźnienia dla potoku z rysunku G.1 dla funkcji X

stałym cyklem zegarowym. Działania dotyczące każdego z etapów potoku są wykonywane w ciągu jednego lub większej liczby cykli zegarowych, a wyniki przesuwane do kolejnego etapu na zboczu cyklu zegarowego. W analizowanym przykładzie ustalamy cykl zegarowy na 20 ns.

Musimy jednak pamiętać, że wiersze tabeli mają odpowiadać zasobom, a nie tylko etapom potoku. Na rysunku G.5b przedstawiono zmodyfikowaną tablicę rezerwacji w celu odzwierciedlenia faktu, że operacje odnoszące się do pamięci wykorzystują tę samą jednostkę funkcjonalną. W tym przypadku jedynym dopuszczalnym opóźnieniem jest to wynoszące 15.

Załóżmy teraz, że zaimplementujemy pamięć podręczną w mikroukładzie i skrócimy czas dostępu do pamięci do jednego cyklu zegarowego. Odpowiednio zmodyfikowaną tablicę rezerwacji pokazano na rysunku G.6a. Przesuwając jedną kopię wzoru tablicy rezerwacji na drugą, łatwo odkrywamy że wektor kolizji wynosi 011010. Powstaje zatem pytanie, jaki harmonogram zapewni najniższe średnie opóźnienie. Jak pokazuje rysunek G.3, istnieje wiele takich możliwości dla dowolnego wektora kolizji.

Wygodnym narzędziem do określania momentu zainicjowania nowego procesu w potoku z jednoczesnym uniknięciem kolizji z niektórymi operacjami znajdującymi się już w potoku jest diagram przejścia stanu wektora kolizji. Procedura tworzenia takiego diagramu jest następująca: