

Na rysunku 18.14 przedstawiono wyjaśnienie sposobu obsługi rozgałęzień, który można opisać następująco:

1. Podczas etapu dekodowania przekazywane są adresy rozgałęzień bezwarunkowych oraz spekulatywnie przekazywane adresy rozgałęzień warunkowych, gdy tylko możliwe jest obliczenie adresu.
2. Jeśli jednostka ALU ustali, że rozgałęzienie nie zostało wykonane, to informacja ta jest zwracana w celu wyczyszczenia pamięci podręcznej rozkazów.
3. Rozkaz ładowania do licznika programu skutkuje przekazaniem adresu rozgałęzienia do pobrania.

Jak można zauważyć, sposób obsługi rozgałęzień w Cortex-M jest znacznie prostszy niż w przypadku Cortex-A oraz wymaga mniej złożonych układów logicznych procesora i mniej jednostek przetwarzania.

## 18.6. PODSTAWOWE POJĘCIA, PYTANIA SPRAWDZAJĄCE I ZADANIA

### Podstawowe pojęcia

antyzależność	superpotokowy	zależność od wyjścia
konflikt zasobów	superskalarny	zależność odczyt-zapis
mikrooperacje	ukończenie poza kolejnością	zależność proceduralna
okno rozkazu	ukończenie w kolejności	zależność przepływu
prawdziwa zależność danych	wycofywanie	zależność zapis-odczyt
przemianowanie rejestrów	wydanie poza kolejnością	zależność zapis-zapis
przewidywanie rozgałęzienia	wydanie rozkazu	zatwierdzanie
równoległość na poziomie maszyny	wydawanie w kolejności	
równoległość na poziomie rozkazu		

### Pytania sprawdzające

- 18.1. Jaka jest zasadnicza cecha superskalarnego podejścia do projektowania procesorów?
- 18.2. Jaka jest różnica między podejściem superskalarnym a superpotokowym?
- 18.3. Co to jest równoległość na poziomie rozkazu?
- 18.4. Krótko zdefiniuj następujące terminy:
  - prawdziwa zależność danych,
  - zależność proceduralna,
  - konflikty zasobów,
  - zależność od wyjścia,
  - antyzależność.

- 18.5. Jaka jest różnica między równoległością na poziomie rozkazu a równoległością na poziomie maszyny?
- 18.6. Wymień i krótko zdefiniuj trzy rodzaje zasad wydawania rozkazów w podejściu superskalarnym.
- 18.7. Jaki jest przeznaczenie okna rozkazu?
- 18.8. Co to jest przemianowanie rejestrów i jaki jest jego cel?
- 18.9. Jakie są kluczowe elementy organizacji procesorów superskalarnych?

## Zadania

- 18.1. Jeśli w procesorze superskalarnym stosowane jest ukończenie poza kolejnością, to wznowienie wykonywania po zakończeniu przetwarzania przzerwania jest skomplikowane, ponieważ warunek wyjątku mógł zostać wykryty jako rozkaz dający wynik poza kolejnością. Program nie może zostać zrestartowany w punkcie rozkazu następującego po rozkazie wyjątku, ponieważ kolejne rozkazy zostały już zakończone, a wykonanie tego spowodowałoby dwukrotne wykonanie tych rozkazów. Zaproponuj mechanizm lub mechanizmy zapobiegające wystąpieniu takiej sytuacji.
- 18.2. Rozważ następującą sekwencję rozkazów zapisanych w postaci: kod operacji, po którym następuje rejestr docelowy, a po nim jeden lub dwa rejestry źródłowe:

```

0   ADD   R3, R1, R2
1   LOAD  R6, [R3]
2   AND   R7, R5, 3
3   ADD   R1, R6, R7
4   SRL   R7, R0, 8
5   OR    R2, R4, R7
6   SUB   R5, R3, R4
7   ADD   R0, R1, 10
8   LOAD  R6, [R5]
9   SUB   R2, R1, R6
10  AND   R3, R7, 15

```

Przyjmij, że wykorzystywany jest potok czteroetapowy obejmujący: pobieranie, dekodowanie/wydawanie, wykonywanie oraz zapis zwrotny. Przyjmij, że wszystkie etapy potoku zajmują jeden cykl zegarowy, z wyjątkiem etapu wykonywania. W przypadku prostych rozkazów arytmetycznych i logicznych wykonywanych na liczbach całkowitych etap wykonywania zajmuje jeden cykl zegarowy, jednak w przypadku rozkazu LOAD odnoszącego się do pamięci etap wykonywania zajmuje pięć cykli.

W przypadku prostego potoku skalarnego, z dozwolonym wykonywaniem poza kolejnością, możliwe jest skonstruowanie poniższej tabeli dotyczącej wykonania pierwszych siedmiu rozkazów:

Rozkaz	Pobranie	Dekodowanie	Wykonanie	Zapis zwrotny
0	0	1	2	3
1	1	2	4	9
2	2	3	5	6
3	3	4	10	11
4	4	5	6	7
5	5	6	8	10
6	6	7	9	12

Wartości umieszczone w kolumnach czterech etapów potoku określają cykl zegarowy, w którym każdy rozkaz rozpoczyna każdą fazę. W analizowanym programie drugi rozkaz ADD (rozkaz 3) zależy od rozkazu LOAD (rozkaz 1) ze względu na jeden z jego argumentów, r6. Ponieważ rozkaz LOAD wymaga pięciu cykli zegarowych, a układ logiczny wydawania napotyka zależny rozkaz ADD po dwóch cyklach, więc układ ten musi opóźnić rozkaz ADD o trzy cykle zegarowe. W obsłudze poza kolejnością procesor może zatrzymać rozkaz 3 w 4. cyklu zegara i następnie przejść do wydania kolejnych trzech niezależnych rozkazów, które wchodzi w etap wykonywania w cyklach 6, 8 i 9. Wykonywanie rozkazu LOAD kończy się w 9. cyklu, a więc wykonywanie zależnego rozkazu ADD może zostać wznowione w 10. cyklu zegarowym.

- Uzupełnij powyższą tabelę.
- Ponownie uzupełnij tabelę, zakładając brak możliwości obsługi poza kolejnością. Jakie są oszczędności wynikające ze stosowania obsługi poza kolejnością?
- Ponownie uzupełnij tabelę, zakładając implementację superskalarną, która na każdym etapie może obsługiwać jednocześnie dwa rozkazy.

### 18.3. Rozważ następujący program w języku asemblera:

```

I1: Move R3, R7           /R3 ← (R7)/
I2: Load R8, (R3)       /R8 ← Memory (R3)/
I3: Add R3, R3, 4        /R3 ← (R3) + 4/
I4: Load R9, (R3)       /R9 ← Memory (R3)/
I5: BLE R8, R9, L3      /Branch if (R9) > (R8)/

```

Ten program zawiera zależności WAW, RAW i WAR. Zademonstruj to.

### 18.4. a. Zidentyfikuj zależności RAW, WAR i WAW w następującej sekwencji rozkazów:

```

I1: R1 = 100
I2: R1 = R2 + R4
I3: R2 = r4 - 25
I4: R4 = R1 + R3
I5: R1 = R1 + 30

```

- Przemianuj rejestry z punktu a, by zapobiec problemom z zależnościami. Zidentyfikuj odniesienia do pierwotnych wartości rejestru, używając indeksu dolnego „a” jako odniesienia do rejestru.

18.5. Rozważ sekwencję wykonywania „wydawanie w kolejności/ukończenie w kolejności” pokazaną na rysunku 18.15.

- Zidentyfikuj najbardziej prawdopodobny powód, dla którego rozkaz I2 nie mógł wejść w etap wykonywania przed czwartym cyklem zegarowym. Czy zastosowanie sekwencji „wydawanie w kolejności/ukończenie poza kolejnością” lub „wydawanie poza kolejnością/ukończenie poza kolejnością” może stanowić rozwiązanie tego problemu? Jeśli tak, to która z nich?
- Zidentyfikuj powód, dla którego rozkaz I6 nie mógł wejść w etap zapisu, aż do 9. cyklu. Czy zastosowanie sekwencji „wydawanie w kolejności/ukończenie poza kolejnością” lub „wydawanie poza kolejnością/ukończenie poza kolejnością” może stanowić rozwiązanie tego problemu? Jeśli tak, to która z nich?

Dekodowanie		Wykonanie			Zapis		Cykl
I1	I2						1
	I2			I1			2
	I2			I1			3
I3	I4		I2				4
I5	I6		I4	I3	I1	I2	5
I5	I6	I5		I3			6
		I5	I6		I3	I4	7
							8
					I5	I6	9

Rysunek 18.15. Kolejność wykonania, wydanie w kolejności, wykonanie w kolejności

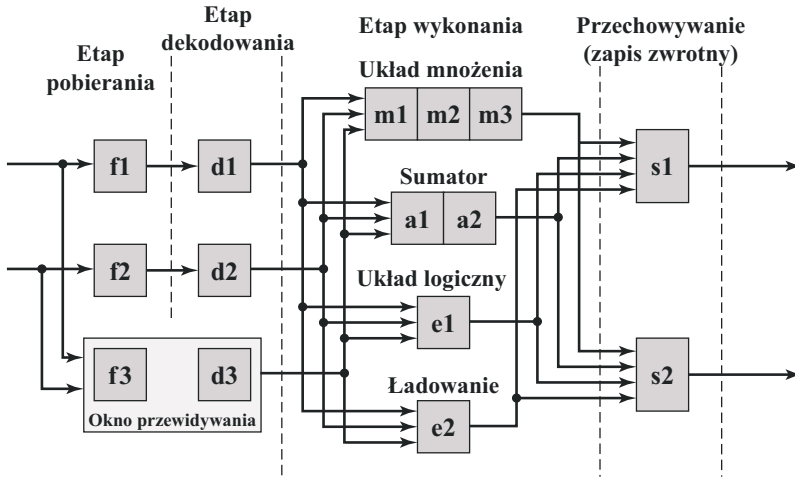
18.6. Rysunek 18.16 przedstawia przykład organizacji procesora superskalarnego. Procesor może wydawać dwa rozkazy na jeden cykl, jeśli nie istnieje konflikt zasobów ani problem wynikający z zależności danych. Zasadniczo istnieją dwa potoki, każdy z czterema etapami przetwarzania (pobieranie, dekodowanie, wykonywanie i przechowywanie). Każdy potok ma własną jednostkę pobierania, dekodowania i przechowywania. Cztery jednostki funkcjonalne (mnożenia, sumowania, jednostka logiczna i jednostka ładowania) są dostępne do użycia na etapie wykonywania i są dynamicznie współdzielone przez dwa potoki. Dwie jednostki przechowywania mogą być dynamicznie wykorzystywane przez dwa potoki w zależności od ich dostępności w danym cyklu zegarowym. Istnieje okno przewidywania (ang. lookahead window) z własnym układem logicznym pobierania i dekodowania. Jest ono używane do przewidywania rozkazów w przypadku wydawania rozkazów poza kolejnością.

Rozważ następujący program przeznaczony do wykonania na tym procesorze:

```

I1: Load R1, A      /R1 ← Memory (A) /
I2: Add R2, R1      /R2 ← (R2) + R(1) /
I3: Add R3, R4      /R3 ← (R3) + R(4) /
I4: Mul R4, R5      /R4 ← (R4) + R(5) /
I5: Comp R6         /R6 ← (R6) /
I6: Mul R6, R7      /R6 ← (R6) × R(7) /

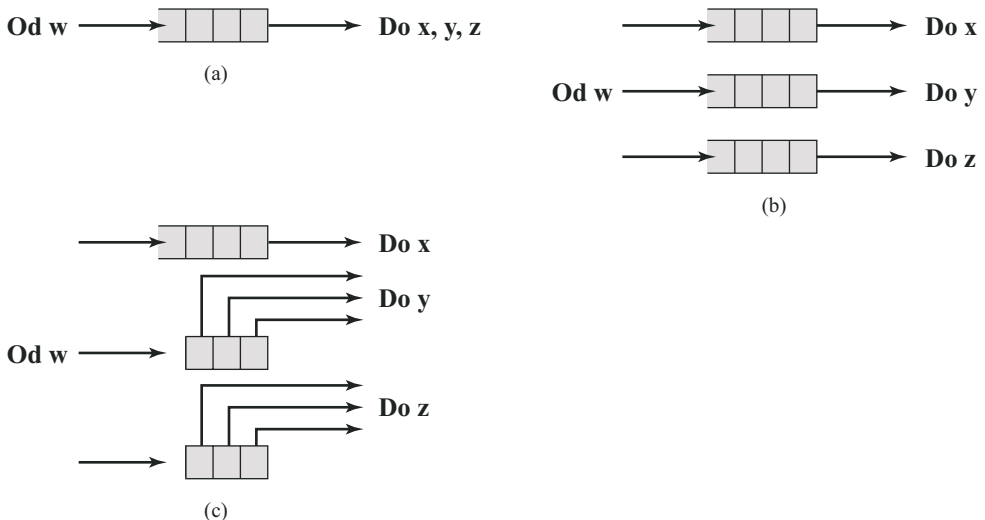
```



Rysunek 18.16. Dwupotokowy procesor superskalarny

- Jakie zależności istnieją w programie?
- Zaprezentuj działanie potoku dla przedstawionego programu na procesorze z rysunku 18.16, stosując zasadę wydawania w kolejności z ukończeniem w kolejności, prezentując wyniki w sposób podobny jak na rysunku 18.3.
- Wykonaj zadanie jeszcze raz, stosując zasadę wydawanie w kolejności z ukończeniem poza kolejnością.
- Wykonaj zadanie jeszcze raz, stosując zasadę wydawanie poza kolejnością z ukończeniem poza kolejnością.

18.7. Rysunek 18.17 został zaczerpnięty z artykułu dotyczącego projektowania superskalarnego. Wyjaśnij trzy części rysunku oraz zdefiniuj w, x, y oraz z.



Rysunek 18.17. Zilustrowanie zadania 18.7