

- 1. PIELINE
 - 1.1. Domanda 1
 - 1.2. Domanda 2
 - 1.3. Domanda 3
 - 1.4. Domanda 4
- 2. TOMASULO
 - 2.1. Domanda 5
 - 2.2. Domanda 6

- **ANS:** Come si vede la mul.d rimane in attesa del registro f2 fino alla fine dello stadio di MEM della I.d. Alla fine di questo stadio di MEM viene attivato il percorso di forwarding verso l'ingresso dello stadio di EX per la mul.d. Il cammino viene rilevato durante la fase di decodifica dell'istruzione di moltiplicazione.

1.4. Domanda 4

Considerando il programma precedente, si consideri che durante la prima iterazione del codice sia scatenata una eccezione da pte della prima delle istruzioni di divisione. In particolare, l'eccezione viene intercettata dal sistema all'ultimo clock cycle della fase di esecuzione (EXE) della divisione. Che conseguenze si osserverebbero sull'esecuzione del codice? motivare 1a risposta

- **ANS:** quando viene intercettata l'eccezione, viene salvato il PC dell'istruzione che l'ha scatenata per poi disabilitare le scritture per l'istruzione stessa e tutte quelle che la seguono, poi viene forzata la routine dell'eccezione nello stage di fetch. Alla fine della gestione della routine dell'eccezione, si ricarica il PC dell'istruzione, che verrà ricominciata e si riprende con il flusso originale delle istruzioni. In questo caso,
 - le due load successive non vengono affette dall'eccezione in quanto hanno terminato prima che si verificasse.
 - Invece, la moltiplicazione subito dopo verrà bloccata e quindi dovrà essere eseguita nuovamente.

2. TOMASULO

2.1. Domanda 5

# iteration		Issue	EXE	MEM	CDB x2	COMMIT x2
1	l.d f3,v3(r1)	1	2m	3	4	5
1	l.d f4,v4(r1)	1	3m	4	5	6
1	div.d f5,f3,f4	2	6d		14	15
1	l.d f1,v1(r1)	2	4m	5	6	15
1	l.d f2,v2(r1)	3	5m	6	7	16
1	mul.d f6,f2,f3	3	8X		14	16
1	div.d f6,f1,f6	4	22d		30	31
1	add.d f5,f6,f5	4	31a		33	34
1	s.d f5,v5(r1)	5	6m			34
1	daddui r1,r1,8	5	6i		7	35
1	daddi r2,r2,-1	6	7i		8	35
1	bnez r2,loop	7	9j			36
2	l.d f3,v3(r1)	8	9m	10	11	36
2	l.d f4,v4(r1)	8	10m	11	12	37
2	div.d f5,f3,f4	9	14d		22	37
2	l.d f1,v1(r1)	9	11m	12	13	38
2	l.d f2,v2(r1)	10	12m	13	15	38
2	mul.d f6,f2,f3	10	16x		22	39
2	div.d f6,f1,f6	11	30d		38	39
2	add.d f5,f6,f5	11	39a		41	42
2	s.d f5,v5(r1)	12	13m			42
2	daddui r1,r1,8	12	13i		15	43
2	daddi r2,r2,-1	13	14i		16	43
2	bnez r2,loop	14	17j			44

Considerando il segmento di codice presentato nella tabella precedente, se assumiamo che il ROB ha una dimensione di 16 elementi, qual è la prima istruzione che dovrebbe stallare durante la esecuzione del programma? motivare la risposta.

- **ANS:** al ciclo di clock 9 abbiamo aggiunto ben 16 elementi all'interno del ROB, quindi ci si aspetta che l'istruzione successiva debba stallare. Questo NON avviene perché al cc 9, le prime due istruzioni hanno già eseguito il commit, liberando due posti nel ROB. Quindi possiamo inserire altre due istruzioni (fino alla mul.d f6, f2, f3). La prima istruzione a stallare sarà la
 - div.d f6, f1, f6 (seconda iterazione)

2.2. Domanda 6

Considerando il segmento di codice presentato nella tabella precedente, se assumiamo che si vorrebbero migliorare le prestazioni del programma duplicando una delle seguenti unità funzionali:

- FP multiplier unit
- FP divider unit
- FP Arithmetic unit

- **ANS:** La scelta migliore è quella di duplicare la unità di divisione, in quanto si osserva dal programma che questa unità crea stalli strutturali che con una ulteriore unità funzionale si potrebbero evitare. In contrapposizione, le unità aritmetiche e di moltiplicazione non ostacolano l'esecuzione del programma.