



1. [4] Spiegare il principio su cui si basa il funzionamento dei sommatore ad anticipazione di riporto. Ricavare la funzioni logiche che definiscono i termini di propagazione e generazione in un sommatore ad anticipazione di riporto a 4 bit.
2. [6] Tradurre in linguaggio Assembly MIPS la seguente funzione, evitando di utilizzare pseudo-istruzioni (si supponga di gestire sempre numeri minori di 2^{32}):

```
int Funzione( int n )
{
    if( n < 1 )
        return( 0 );
    else
        return( 1 + n * Funzione(n-1) );
}
```

3. [3] Tradurre in Assembly nativo e in linguaggio macchina MIPS le seguenti pseudo-istruzioni:

```
li $2, 0x12345678
blti $6, 5, +12      # blti = branch on less than immediate
```
4. [7] Progettare una macchina a stati finiti di Moore che implementa un contatore da -2 a +2. Il contatore riceve 3 ingressi binari: "INCR", che quando va a "1" incrementa il contatore di 1, "DECR" che lo decrementa di 1, e "RESET" che lo azzerà. Quando il contatore raggiunge +2, i successivi incrementi vengono ignorati, e lo stesso per i decrementi quando vale -2. Si supponga, per semplicità, che solo un ingresso alla volta possa andare a "1". Il contatore presenta 2 uscite binarie: una va a "1" quando il contatore è arrivato a +2, l'altra quando è a -2. Determinare STG, STT, STT codificata e determinare le funzioni logiche di uscita e stato prossimo.
5. [6] Si consideri una memoria cache 4-associativa, che serve un processore con bus dati di 16 bit e bus indirizzi di 24 bit. La capacità totale della cache è di 64 kbyte e la dimensione di blocco è di 16 parole.
Dimensionare la cache, definendo come viene utilizzata la parola d'indirizzo di memoria. Disegnare lo schema circuitale di tale memoria, esplicitando le dimensioni di tutti i bus. Determinare quali bytes della memoria cache possano venire modificati eseguendo l'istruzione:

```
sw $s1, 0x1020FF
```
6. [4] Data una memoria RAM dinamica di 1 M x 1 bit, disegnarne la struttura circuitale e calcolare il periodo massimo di refresh, supponendo il tempo di scarica delle celle di memoria pari a 25,6 msec.
7. [3] In un calcolatore, l'introduzione di una memoria cache ha portato ad un incremento globale della velocità di 2 volte, con un hit-rate del 75%.
Di quante volte è più veloce la cache introdotta rispetto alla memoria principale?
Quanto aumenterebbe la velocità del calcolatore se la velocità della cache fosse infinita?

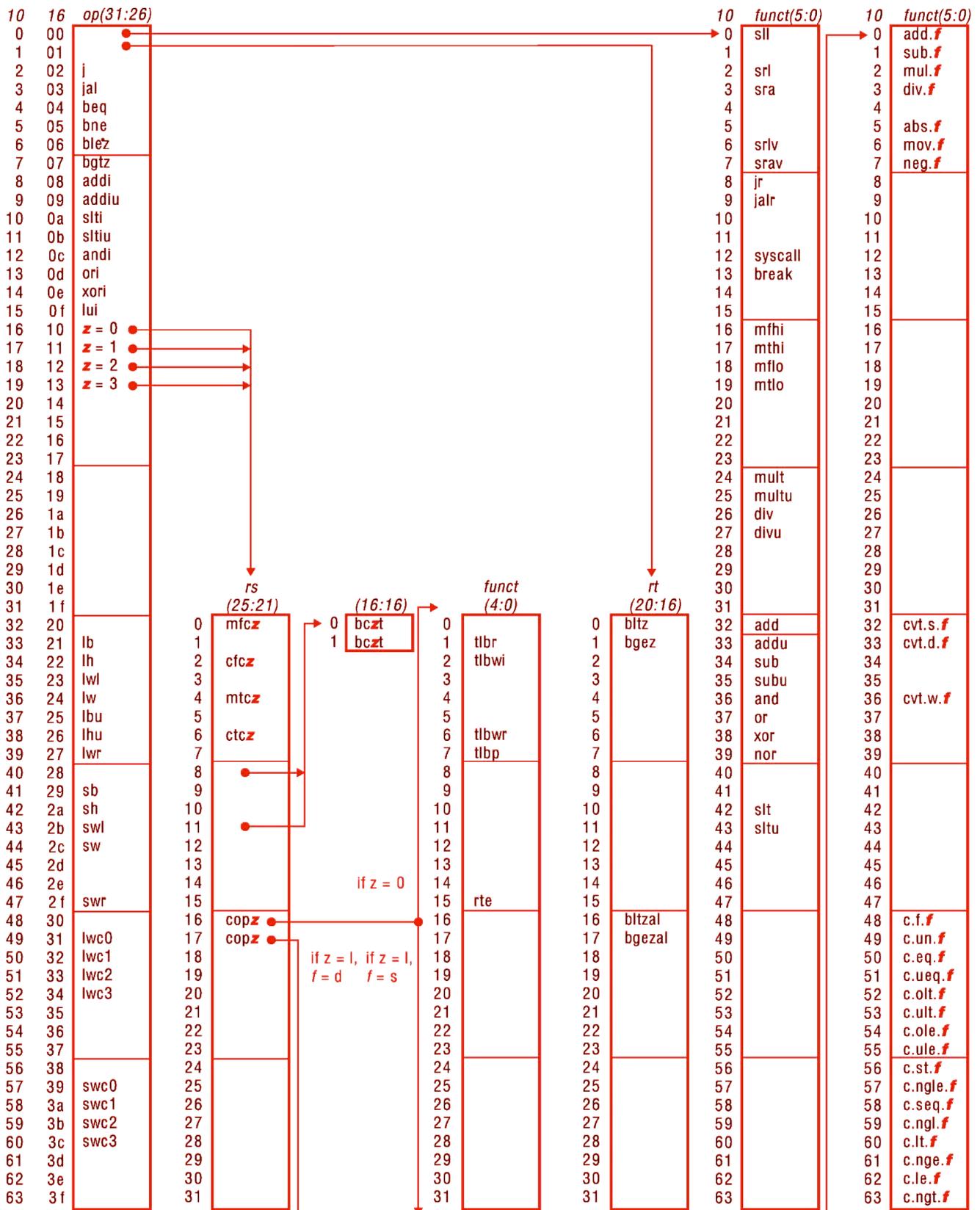


FIGURE A.19 MIPS opcode map. The values of each field are shown to its left. The first column shows the values in base 10 and the second shows base 16 for the op field (bits 31 to 26) in the third column. This op field completely specifies the MIPS operation except for 6 op values: 0, 1, 16, 17, 18, and 19. These operations are determined by other fields, identified by pointers. The last field (funct) uses “f” to mean “s” if rs = 16 and op = 17 or “d” if rs = 17 and op = 17. The second field (rs) uses “z” to mean “0”, “1”, “2”, or “3” if op = 16, 17, 18, or 19, respectively. If rs = 16, the operation is specified elsewhere: if z = 0, the operations are specified in the fourth field (bits 4 to 0); if z = 1, then the operations are in the last field with f = s. If rs = 17 and z = 1, then the operations are in the last field with f = d. (page A-54)