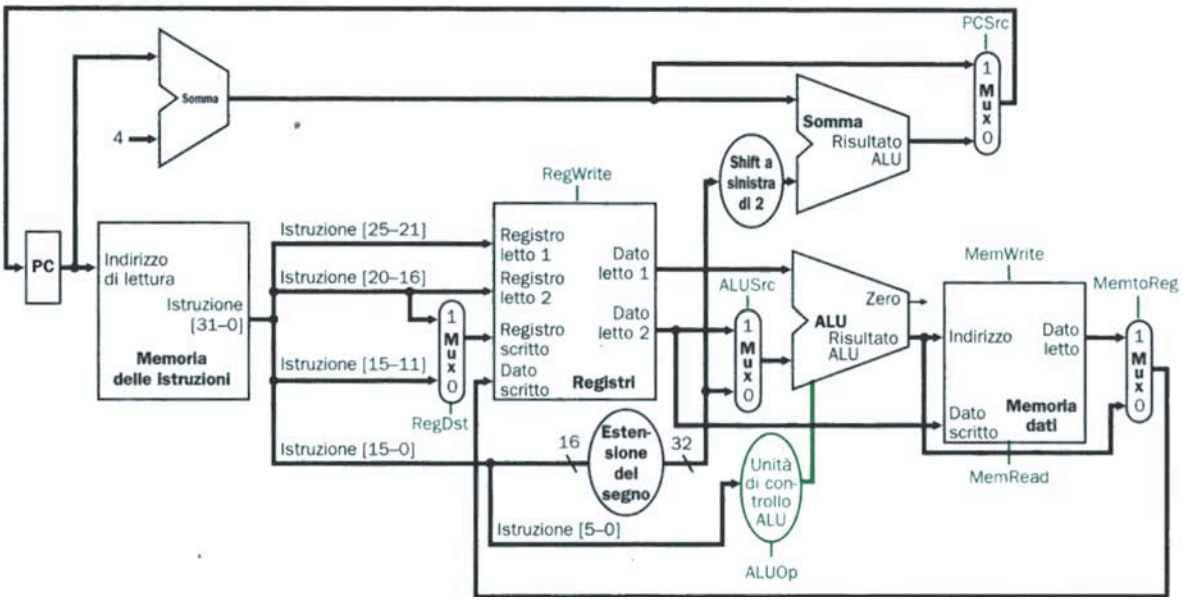
1. [5] Si progetti un circuito caratterizzato da un ingresso a 4 bit che rappresenta un numero binario A, e da un'uscita che vale '1' se: **A è divisibile per 3**, oppure se **7 < A < 12**; altrimenti l'uscita vale '0'.
   - Determinare la tabella di verità della funzione logica di uscita;
   - scrivere la funzione nella forma canonica più adatta;
   - semplificarla mediante mappa di Karnaugh e poi, se possibile, semplificarla ulteriormente mediante passaggi algebrici;

2. [4] Evidenziare, nel seguente schema di CPU, i valori all'ingresso ed all'uscita di ogni ALU ed i valori di ogni segnale di controllo, supponendo che la CPU stia eseguendo l'istruzione: **`0x100: beq $1, $2, +24`**



3. [8] Si sintetizzi una macchina a stati finiti di Moore che accetti in ingresso un carattere binario (0 o 1) e sia caratterizzata da un'uscita binaria, la quale vale "1" quando **il bit in ingresso è uguale ai due bit precedenti**; in tutti gli altri casi vale "0". Allo stato iniziale, la macchina assume come bit precedenti "00".
   Si determinino STG, STT, STT codificata e struttura circuitale del sistema completo, avendo cura di semplificare il più possibile le funzioni prima di tradurle in circuito.

4. [6] Si progetti (esplicitando le dimensioni di tutti i campi) una memoria cache ad otto vie, associata ad un processore con bus dati di 16 bit e bus indirizzi di 24 bit. La capacità totale della cache sia di 1 Mbyte e la dimensione del blocco di 16 parole. Se ne rappresenti lo schema circuitale dettagliato e si calcoli il valore (decimale) di tutti i campi di indirizzamento relativi all'indirizzo: **`0xABCDEF`**.

5. [4] In un calcolatore, l'introduzione di una memoria cache ha portato ad un incremento globale della velocità di 2 volte, con un hit-rate dell' 80%. Con tale hit-rate, quanto sarebbe l'incremento di velocità massimo raggiungibile?

6. [4] Si consideri un codice di controllo errori nel quale, ad ogni tre bit del messaggio originale, viene aggiunto un ulteriore bit, calcolato come XOR dei tre bit del messaggio. Calcolare il costo del codice, la sua distanza minima, la capacità di rivelazione e la capacità di correzione.

7. [5] Si traducano le seguenti pseudoistruzioni: **a)** in Assembly MIPS nativo e **b)** in linguaggio macchina MIPS (specificando dimensione in bit e valore dei campi di ogni istruzione):
   a) **`divi $1, $10, 100  # divide immediate`**
   b) **`bgei $1, +10, 100  # branch on greater than or equal to immediate`**

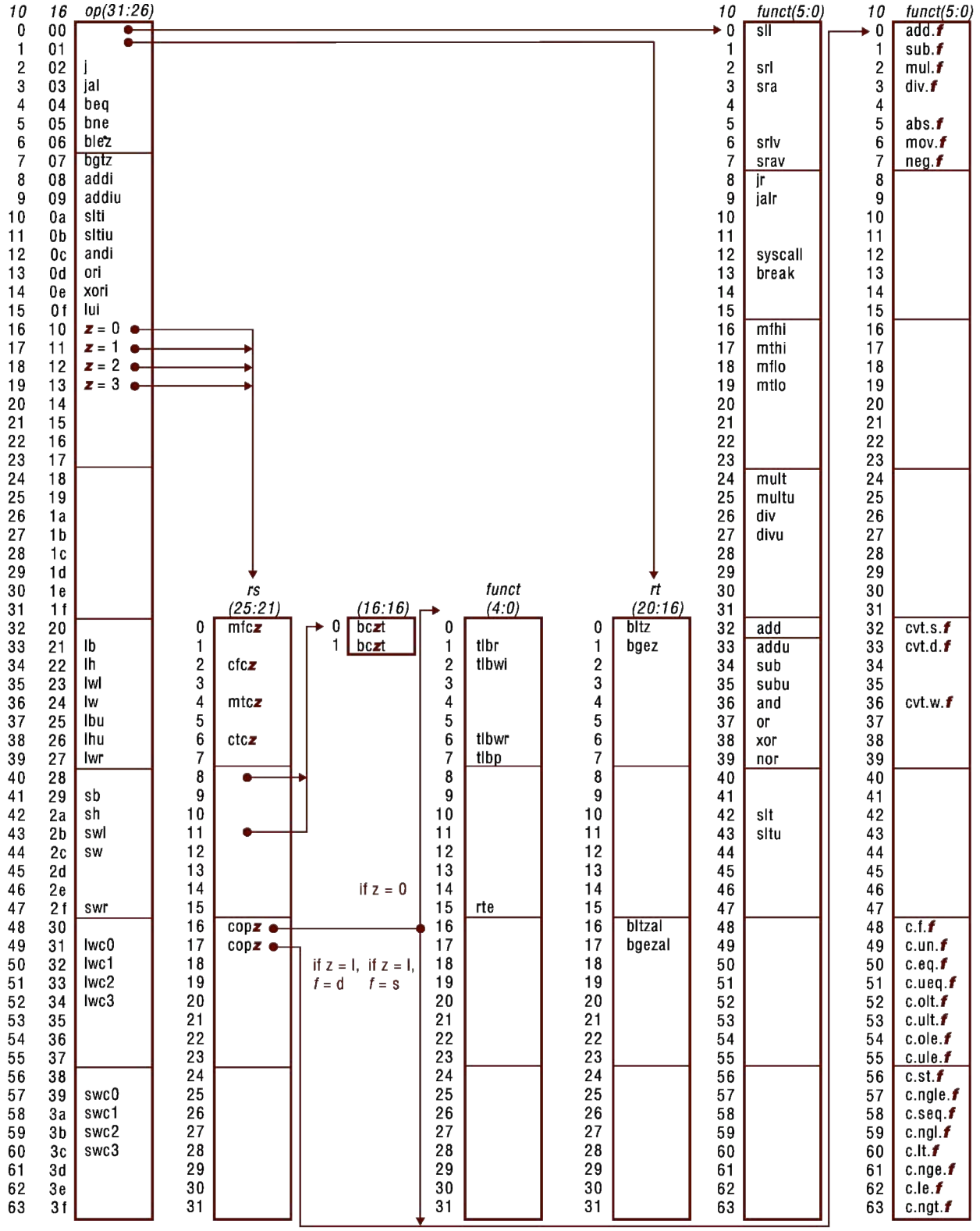


FIGURE A.19 MIPS opcode map. The values of each field are shown to its left. The first column shows the values in base 10 and the second shows base 16 for the op field (bits 31 to 26) in the third column. This op field completely specifies the MIPS operation except for 6 op values: 0, 1, 16, 17, 18, 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)*