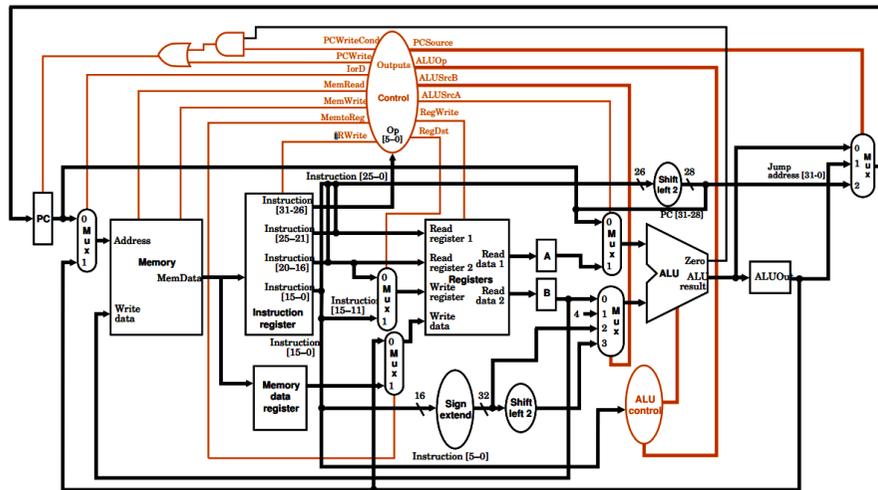




1. [6] Nella CPU in figura, si supponga di voler aggiungere all'Instruction Set l'istruzione *branch on not equal* (**bne**, formato I, Opcode: 5).
Progettare le modifiche architetturali alla presente CPU (datapath) e le modifiche alla macchina a stati finiti dell'unità di controllo (specificando i valori degli eventuali segnali di controllo e specificando la modifica dello STG dell'unità di controllo).



2. [4] Elencare e descrivere i criteri più comuni utilizzati nelle memorie cache a più vie per la scelta del banco in cui scrivere un nuovo blocco.
3. [4] Descrivere come viene gestita un'eccezione in una CPU, esplicitando i componenti hardware utilizzati e spiegandone il funzionamento.
4. [4] Si consideri un codice di controllo errori che, ad ogni coppia di bit, aggiunge la stessa coppia e poi la coppia negata (ad es.: 01 → 0110). Elencare le parole di codice valide, calcolare il costo del codice, la distanza minima e le capacità di rivelazione e di correzione su una parola.

5. [5] Un elaboratore dotato di memoria virtuale dispone di una memoria fisica di 256 MB e gestisce uno spazio di memoria virtuale di 16 GB. Considerato che la dimensione delle pagine è di 4 kB e l'ampiezza di parola è di 32 bit, calcolare:
a) la dimensione della Page Table (supponendo che ogni linea di page table occupi una parola di memoria);
b) Virtual Page Number e Page Offset dell'indirizzo: **0x123456780**.
6. [4] Descrivere la struttura e il funzionamento dell'unità di propagazione ("forwarding unit") della CPU pipeline MIPS, identificando tutti gli ingressi e le uscite dell'unità e definendo esattamente il suo algoritmo di funzionamento.
7. [6] Un programma Assembly MIPS ha bisogno di allocare staticamente un array di 100 interi e inizializzarlo con i numeri da 99 a 0, in ordine. Deve poi allocare dinamicamente un altro, dopo aver chiesto all'utente (da terminale) il numero di elementi che deve contenere, che devono essere tutti inizializzati a 25. Scrivere il codice che costruisce tali array.

System calls

	codice (\$v0)	argomenti	risultato
print_int	1	\$a0	
print_float	2	\$\$f12	
print_double	3	\$\$f12	
print_string	4	\$a0	
read_int	5		\$v0
read_float	6		\$\$f0
read_double	7		\$\$f0
read_string	8	\$a0, \$a1	
sbrk	9	\$a0	\$v0
exit	10		

Registri MIPS

0	zero	24-25	t8-t9
1	at	26-27	k0-k1
2-3	v0-v1	28	Gp
4-7	a0-a3	29	Sp
8-15	t0-t7	30	s8
16-23	s0-s7	31	Ra