

## 1. CALCOLATORI ELETTRONICI – PROVA II – 2/2/2017 – II FACOLTA' DI INGEGNERIA (CESENA)

COGNOME	NOME	MATRICOLA	PARI/DISPARI	POSTO
			Non interessa	

**TESTO - Tempo disponibile: 55 minuti – Il disordine nella soluzione porta a un abbassamento del voto.**

Si vuole estendere l'ISA del DLX visto a lezione con istruzioni di chiamata e ritorno da procedura che utilizzano uno STACK localizzato in memoria principale (cioè indirizzato tramite MAR). **In particolare in questo compito ci occupiamo dell'istruzione di ritorno da procedura.**

### Prime domande introduttive al compito (Punti 7)

Per cominciare si risponda ai seguenti quesiti:

1. Dove viene salvato l'indirizzo di ritorno da chiamata a procedura nel DLX standard visto a lezione?
2. Qual è quindi l'istruzione con cui si esegue il ritorno da chiamata a procedura nell'ISA del DLX?
3. Di che tipo sarà questa istruzione? (R, I o J)? Se ne disegni il formato mettendo in evidenza i campi necessari.
4. Facendo riferimento all'architettura del DLX vista a lezione si dica come i bit del registro IR (Instruction Register) della U.d.C. vengono collegati ai decoder del *register file* nel caso di questa istruzione appena illustrata.

### Progetto della nuova istruzione con stack pointer in R30 (Punti 26).

Si assegna al registro **R30** la funzione di Stack Pointer (SP) e si adottano le seguenti convenzioni:

- Lo stack cresce verso gli indirizzi decrescenti
- Lo SP punta all'ultimo byte occupato (in caso di dati salvati in memoria in formato little endian, questo è il byte meno significativo dell'ultima word salvata sullo stack; questo indirizzo si chiama "**top of stack**" cioè cima dello stack).

Inoltre si assume che SP (cioè R30) sia stato inizializzato con un valore multiplo di 4 (si assume cioè che gli elementi dello stack siano allineati).

Tra le istruzioni che utilizzano lo stack si consideri, come preannunciato, la istruzione di **ritorno da procedura ipotizzando che l'indirizzo di ritorno si trovi in cima allo stack (top of stack)**. Nei programmi in assembler per questo DLX modificato la nuova istruzione verrà scritta come segue:

- **RET\_STACK (cioè RETURN VIA\_STACK)**

Questa istruzione può essere realizzata dal data path ad esempio con le seguenti operazioni RTL:

**PC ← M[R30]**

**R30 ← R30+4**

1. Si ipotizzi che il codice operativo della istruzione **RET\_STACK** sia **31H** e si proponga una buona *codifica binaria della nuova istruzione* scegliendo uno dei 3 formati R, I, J delle istruzioni del DLX. Quanti e quali operandi sorgente su register file ha la nuova istruzione nel formato proposto? Quanti e quali operandi destinazione (sempre su register file)? **(punti 4)**
2. Con riferimento alla *codifica dell'istruzione* scelta e con riferimento al datapath del DLX sequenziale visto a lezione, si disegni ora il diagramma degli stati che controlla l'esecuzione dell'istruzione assegnata utilizzando il minor numero di stati possibile, inserendo anche gli stati necessari alle fasi di fetch e decodifica. Quanti periodi di clock sono necessari per eseguire l'istruzione appena progettata nel caso in cui l'accesso alla memoria richieda 2 stati di wait? **(Punti 13)**
3. Si chiede ora di scrivere nell'ISA del DLX "standard" la sequenza di istruzioni necessaria a eseguire la stessa funzione svolta dalla istruzione appena progettata e si calcoli il relativo speed up rispetto all'esecuzione della stessa funzione con il DLX standard. Si motivi questo speed up **(punti 7)**.
4. Si chiede ora di indicare come si potrebbe modificare l'architettura del datapath se si volesse dotare la nostra CPU di stack senza utilizzare un registro del register file come Stack Pointer. In questo caso come e perchè si semplificherebbe il formato dell'istruzione? **(Punti 2)**.