1. CALCOLATORI ELETTRONICI – PROVA II – 21/2/2017 – II FACOLTA' DI INGEGNERIA (CESENA)

COGNOME	NOME	MATRICOLA	PARI/DISPARI	POSTO
			Non interessa	

TESTO - Tempo disponibile: 55 minuti – Il disordine porta a un abbassamento del voto.
Si vuole estendere l'ISA del DLX visto a lezione con istruzioni che gestiscono procedure e interrupt attraverso uno STACK localizzato in memoria principale (cioè indirizzato tramite MAR). **In particolare in questo compito ci occupiamo dell'istruzione di ritorno da interrupt.**

Prime domande introduttive al compito (Punti 12)
Per cominciare si risponda ai seguenti quesiti:
1. Dove viene salvato l'indirizzo di ritorno da interrupt nel DLX standard visto a lezione?
2. Qual è l'istruzione con cui si esegue il ritorno da interrupt 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.
   Si risponda ora ai seguenti tre quesiti del tutto generali:
4. In una istruzione di tipo R si può fare riferimento al PC nei campi Rs1, RS2 o Rdest? Perché? (*3 punti*)
5. Altrimenti come si potrebbe creare una istruzione che fa riferimento al PC?
6. Qual è la funzione del Flag IEN che si trova all'interno della UdC?

Progetto della nuova istruzione con stack pointer in R30 (Punti 21).
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 interrupt 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:
- **RFE_STACK (cioè RETURN FROM EXCEPTION VIA_STACK)**

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

PC ← M[R30]
R30 ← R30+4
IEN ← 1 (si ipotizzi che IEN sia nell'unità di controllo)

1. Si ipotizzi che il codice operativo della istruzione **RFE_STACK** sia **23H** e si proponga una *codifica binaria della nuova istruzione* che non richieda la modifica delle connessioni standard tra IR e decoder del Register File, scegliendo quindi uno dei 2 formati R o I delle istruzioni del DLX. Quanti e quali operandi sorgente su register file ha la nuova istruzione nel formato binario 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 11)**

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. Quale registro di transito conviene utilizzare? Si calcoli infine lo speed up della RFE_STACK rispetto all'esecuzione della stessa funzione con il DLX standard. Si motivi questo speed up **(punti 6).**