

Progetto Giano

## Custom Dummy Integrations

C.Baffa, E.Giani

Memo, 5 Aprile 2011

Il presente memo documenta la possibilità di specificare nel dettaglio il tipo di operazioni eseguite dall'elettronica di sequenza quando vengono attivate le *dummy-acquisitions*. Tale possibilità è stata implementata come operazione di test avanzato e quindi è funzionalmente completa, ma è abbastanza semplice nell'interfaccia.

### Funzionamento

Attivando la voce "Load Dummy File" nell'interfaccia di Guilab, il file `data/resources/dummycustom.txt` viene analizzato e convertito in istruzioni base del generatore di sequenze. La lista di tali istruzioni viene scritta nel file `dummyobj.txt` che viene inviato al programma di gestione dell'elettronica IronGolem/Server104. Da quel momento le integrazioni dummy utilizzeranno la sequenza specificata nel file di partenza. Per ricaricare un diverso file, basta disabilitare e riabilitare l'opzione. Per tornare al modo di default, occorre disabilitare l'opzione e eseguire una qualunque integrazione.

Il vincolo che il file di partenza abbia un nome prefissato non è un vincolo stringente come potrebbe apparire in un primo momento, perché è sempre possibile costruire numerosi file con nomi descrittivi e creare un *logical-link* al nome `dummycustom.txt`.

### Formato del file in ingresso.

Il file in ingresso (`dummycustom.txt`) è un file di testo che ammette sia commenti, sia linee di test. Il carattere che segnala l'inizio dei commenti è il cancelletto (`#`). Tutto ciò che segue tale carattere viene ignorato. Le righe valide contengono come primo argomento la sigla (2 caratteri) dell'operazione desiderata seguita da nessuno, uno o due argomenti numerici. Un esempio può essere:

```
# questo è un commento  
SR 16  
RD 64 63 # altro commento  
WT 1  
RS
```

### Istruzioni di programma

Ogni comando del generatore di sequenze è composto da un intero a 16 bit. I primi due bit dell'istruzione indicano il tipo di operazione. I restanti 14 bit sono considerati in modo diverso a seconda del tipo di

operazione. I comandi possibili sono:

Mnemonic	Argomento 1	Argomento 2	Significato
SR	Ripetizione < 128	//	Short Reset
DR	Ripetizione < 128	//	Dry Reset
RR	Read Clock < 128	Read Delay < 128	Reset e Read
WT	Ritardo < 16300	//	Attesa (Wait)
RD	Read Clock < 128	Read Delay < 128	Read
RS	//	//	Restart

### Dettaglio degli argomenti delle varie istruzioni

**SHORT RESET:** Esegue un reset dell'elettronica. I 14 bit di argomento sono interpretati come due gruppi di 7 bit. Il gruppo di bit da 0 a 6 non sono significativi, mentre quelli da 7 a 13 sono tutti 0.

**RESET-READ:** Esegue una lettura con reset. Per le operazioni di Reset-Read e bit da 7 a 13 forniscono la durata del semiperiodo di clock secondo la formula:

$$(Val+16)*62.5\text{ ns}; Val < 2^7$$

La commutazione del pixel si ha sempre sul fronte di salita del clock di pixel, per cui il tempo di scansione del pixel è esattamente il doppio di questo valore. Il valore minimo si ha per  $Val = 0$  e corrisponde a 2  $\mu$ s.

Il gruppo dei bit da 0 a 6 rappresenta invece il ritardo di campionamento rispetto al clock (in particolare il ritardo rispetto al fronte di discesa del clock di pixel), calcolato usando la formula precedente. Se questo valore risulta maggiore di quello ottenuto con i 7 bit piú significativi, otteniamo una lettura senza conversione.

**READ:** Esegue una lettura dell'array. Ha due argomenti che hanno lo stesso formato e lo stesso significato dell'istruzione **RESET-READ**

**ATTESA (Wait):** Nell'istruzione di Attesa, il campo dei 14 bit rimanenti costituisce il tempo di integrazione, il cui valore è espresso dalla formula:

$$(Val+1)*10\text{ ms}; Val < 2^{14}$$

Il tempo minimo di integrazione (tutti i 14 bit uguali a zero) è 10 ms.

**RESTART:** Per l'istruzione Restart i 14 bit restanti non sono significativi.