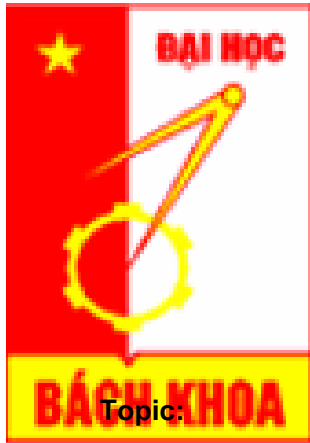


**Hanoi University of Science and Technology**

**ISTITUTO DI ELETTRONICA - TELECOMUNICAZIONI**

**ESRC LAB**



Topic:

**GIOCO DEL SERPENTE su Kit DE1**

**Gruppi di studenti:**

**Gruppo 1 - K53:**

**Vu Quang Trong**

**Do Son Tung**

**Hanoi 8/2011**

## 1. introdurre

### 1.1. filo

Dopo il completamento della pratica di laboratorio su Kit Altera DE1, abbiamo continuare a sviluppare le proprie competenze e applicare nella pratica di progettazione, che viene distribuito un sistema completo su Kit DE1 con il tema:

## **"Utilizzando il kit Altera DE1 per creare un gioco Snake**

**i giocatori con un'interfaccia grafica, comunicare con i giocatori tramite la tastiera**

**PS2 e monitor VGA. "**

### **1.2. Membri e posti di lavoro assegnare**

Immagine andate qui

Immagine andate qui

Vu Quang Trong

Do Son Tung

(Leader)

0973.750.337

0168.9.929.537

[vuquangtrong@gmail.com](mailto:vuquangtrong@gmail.com)

[tungmontaint@gmail.com](mailto:tungmontaint@gmail.com)

Mappatura del tema generale.

Logico stato del sistema di blocco.

Imparare la connessione PS2.

blocco di controllo solido.

Imparare e driver VGA.

blocchi di display grafico.

Imparare AudioCodec e controllo IC

blocchi di testo visualizzato.

Il volume del suono.

E blocchi altri accessori necessari.

### **1.3. Requisiti di argomento**

#### **1.3.1. Requisiti funzionali**

- hardware
  - o Il gioco gira interamente su Kit DE1
  - o Prendi il controllo della tastiera PS2

- o Mostra sulla risoluzione dello schermo VGA di 640 x 480

- o Suono attraverso un altoparlante 2.0 con codec audio WM8731 IC ha

- manioca su Kit DE1

- software

- o Quartus II 9.1 SP2 utilizzando

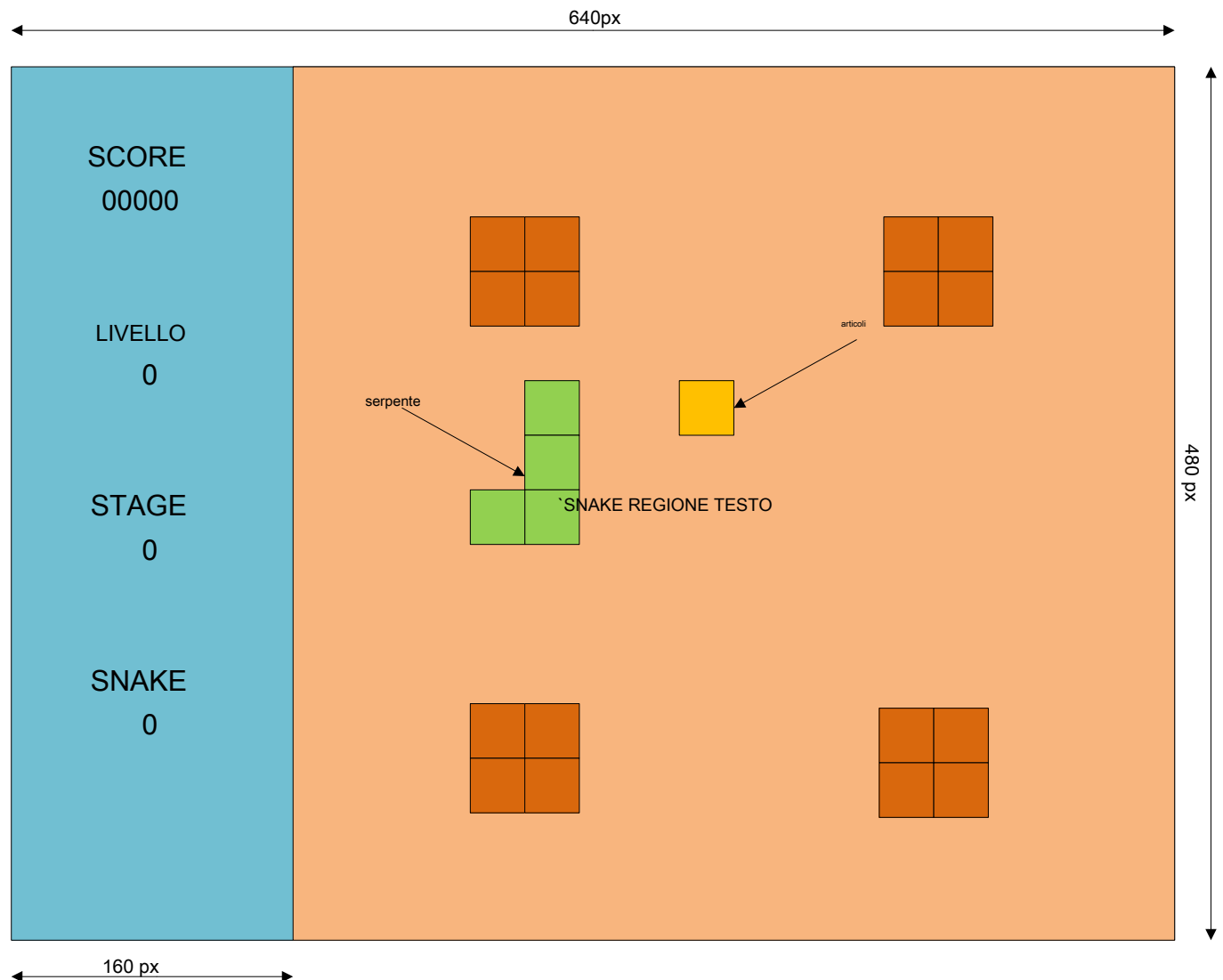
- o Pure linguaggio VHDL

- o Il gioco ha tutti gli elementi per diventare un gioco completo:

- Il giocatore controlla serpente con 4 pulsanti direzionali sul tastiera.
- I tasti funzionano come pausa, la gente SELEZIONA digita gioco MENU nella manipolazione del gioco.
- Lungo serpente durante l'alimentazione, o hanno altre interazioni seconda esche gusto.
- Precetti moriranno se il serpente colpito le pareti o la coda che, a questo punto si avrà un nuovo thread di riproduzione solidi.
- Condizioni per giocare di nuovo se e solo se il numero di riproduzioni di si maggiore di 0, l'originale, i giochi saranno 3, possono se mangiare aumento esche LIVE\_UP, e diminuiscono quando il serpente morto. Se il numero di riproduzioni la schiena è 0, il gioco finirà.
- I diversi livelli di difficoltà, che è la velocità del serpente aumenterà attraverso un certo numero di volte Predator, seguita dalla schermata di gioco ci vari ostacoli.
- Il punteggio per il gioco, il punteggio viene calcolato a livello attuale e esche che i serpenti mangiano.
- Salva giocatori di nome quando i giocatori raggiungono un punto il punteggio nella top 5

### 1.3.2. requisiti non funzionali

- Tempo di risposta meno sequenze di tasti per aiutare i giocatori di navigazione  
i controlli facili, velocità di stampa circa 4 volte / sec => tempo  
250ms risposta
- Visualizzato sullo schermo 640x480 VGA con 8 colori di base, utilizzare 3  
bit per il colore.
- Frequenza Lo schermo è abbastanza grande per garantire la visualizzazione regolare  
che prendiamo è di 60 Hz di frequenza di scansione.
- leggi di controllo vengono applicate nel modo seguente:  
  
o I serpenti non possono tornare indietro nella direzione del viaggio, vale a dire  
  
come se andando avanti, quindi il pulsante non funziona a ritroso,  
  
simile quando i denti sono a sinistra, a destra, o verso il basso.
- Telaio di giocatori:



schermata di gioco è suddivisa in due riquadri principali:

- Informazioni sul display cornice del giocatore:
  - o Tra cui punteggio, livello, stage e numero di giocatori rimasti solidi.
  - o 160x480 dimensioni.
  - o Il testo viene visualizzato con dimensioni 32x16.
- parte di struttura dell'esposizione si gioca:
  - o dimensioni 480x480

o Serpenti, preda, e la parete è stato costruito da blocchi di 16x16 quadrato rimontaggio.

o Questa sezione può essere per visualizzare un messaggio quando è necessario.

## 1.4. hardware consigliato

### 1.4.1. kit DE1

#### 1.4.1.1. L'introduzione di KIT DE1

KIT DE1 è un vero e proprio prodotto di Altera. Muc atterraggio dello sviluppatore

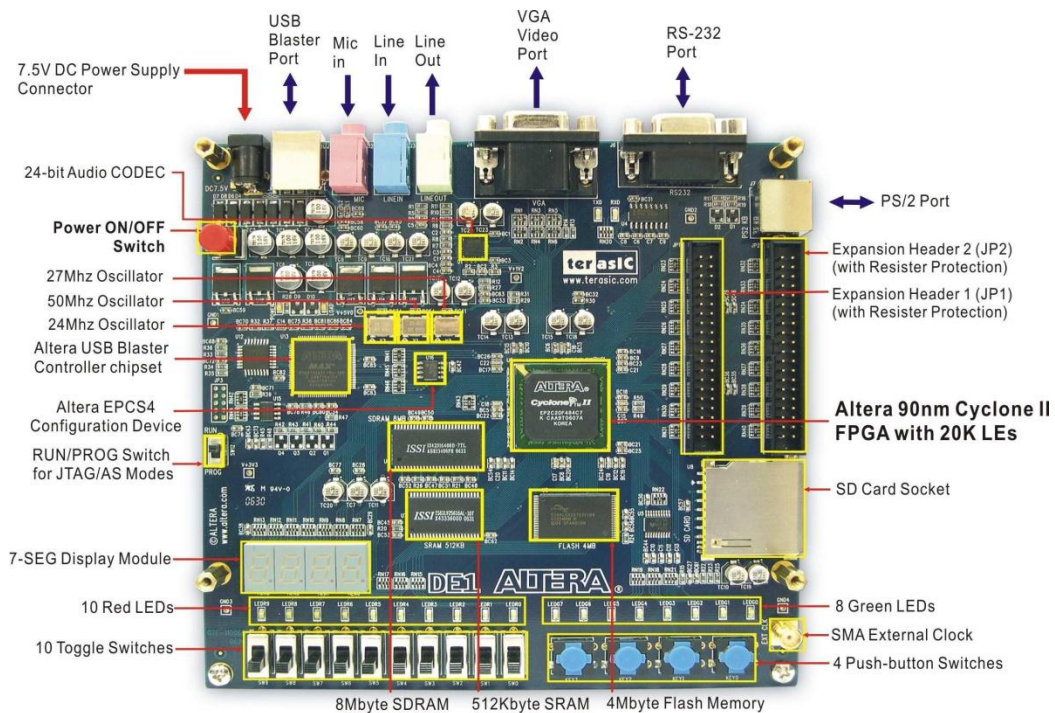
KIT DE1 durante la creazione di uno strumento che offre il servizio ideale per design avanzato in alcuni settori come la multimedialità, la conservazione, rete ...

KIT DE1 da usare, abbiamo bisogno di connettersi a un computer che esegue il software Microsoft Windows.

Figure 1.1 shows a photograph of the DE1 package.



#### 1.4.1.2. componenti Kit su DE1



- Dispositivo altera Cyclone® II 2C20 FPGA
- Altera dispositivo di configurazione seriale - EPCS4
- Blaster USB (on board) per la programmazione e API controllo utente; entrambi

### JTAG e Active Serial

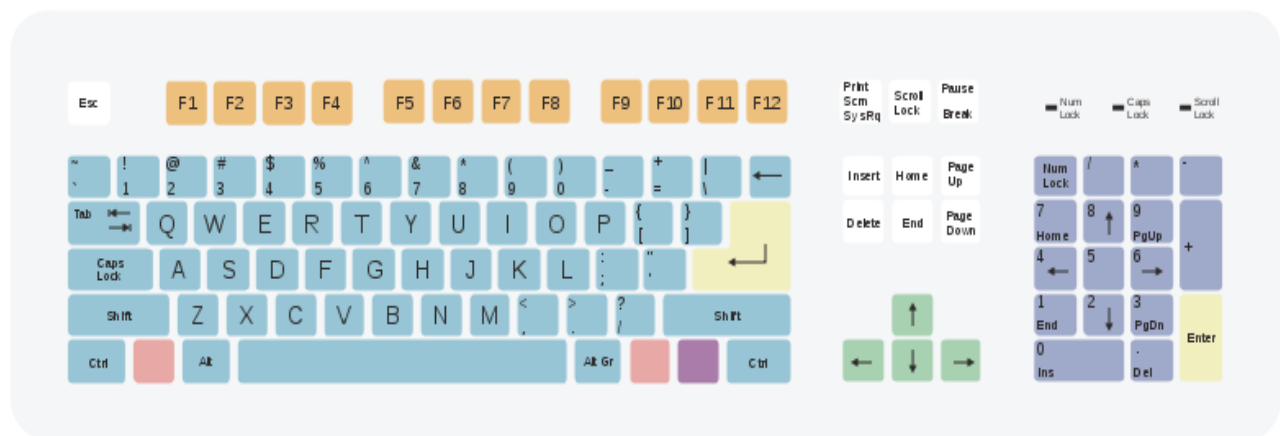
(AS) modalità di programmazione sono supportati

- 512 Kbyte SRAM
- 8 Mbyte SDRAM
- memoria da 4 Mbyte Flash
- Vano scheda SD
- 4 interruttori a pulsante
- 10 interruttori basculanti
- 10 utente LED rosso
- 8 LED utente reen



- 50 MHz oscillatore, 27 MHz e 24 MHz oscillatore oscillatore di clock
- fonti
- Qualità CD CODEC audio a 24 bit con il line-in, line-out, microfono-e jack di ingresso
- DAC VGA (rete resistiva 4-bit) con connettore VGA-out
- RS 232-ricetrasmittitore e connettore 9 poli
- Mouse PS / 2 Connettore / Tastiera
- Due collettori 40 poli espansione con resistenza Protec
- Alimentato da HOAC un adattatore DC 7.5V o un cavo USB

### 1.4.2. PS2 Keyboard



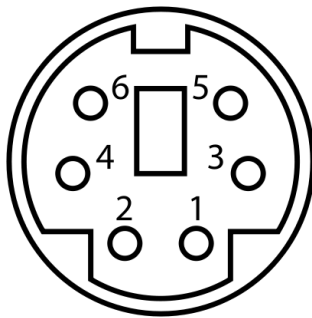
- |   |   |  |
|---|---|--|
|  Typewriter keys |  Function keys       |  Enter keys |
|  System keys     |  Numeric keypad      |  Other      |
|  Application key |  Cursor control keys |  |

Con la nostra gamma progettato questo progetto, interessati solo i pulsanti Moving cosa cane: su, giù, sinistra, destra. Due tasti funzione mettere in pausa e le opzioni: Esc, Enter.Co possono sviluppare più altri nodi requisiti per ciascun soggetto.

I dettagli su come ricevere e trasmettere dati chiave dalla tastiera saranno elencati in sezione illustra il sistema.

## pinout PS2

Utilizzare connettore PS2 standard per collegare una tastiera con KIT DE1



Pin 1 + DATI I dati

Pin 2 Non collegato Non collegato \*

pin GND 3 Gr

Batteria Vcc +5 V DC 4 a 275 mA

Pin 5 + CLK

Pin 6 Non collegato Non collegato \*\*

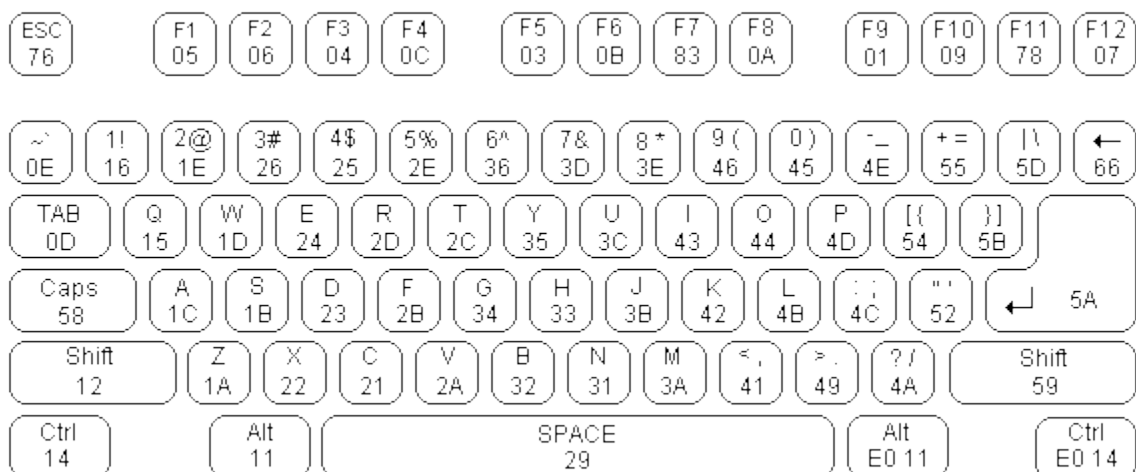
Non standard PS2 ha qualche altro standard di connessione comune è USB e senza fili (wireless). Ambito di temi progettuali legati solo con la PS2 Dovremmo introdurre e conoscere gli altri due standard.

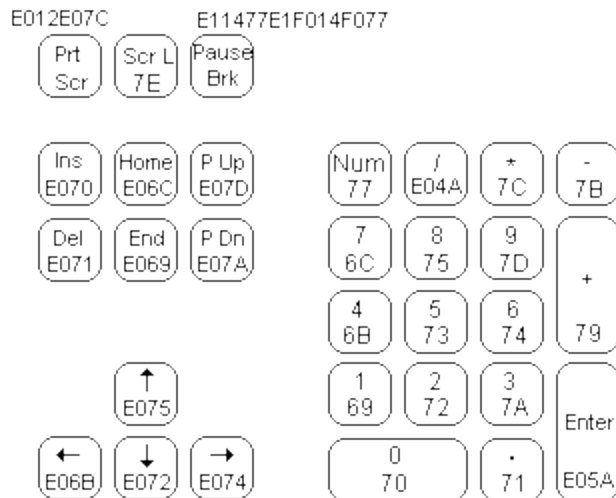
connettore di interfaccia PS2 è un 6-pin MINI DIN.

### 2.1.2.3. codice di scansione

Una tastiera comprende una matrice chiave e un microprocessore per

controllare il funzionamento dei tasti e inviare *codice di scansione* di conseguenza.





#### Il funzionamento della tastiera:

- Quando si preme un tasto, il primo *il codice make* i tasti vengono trasmessi.
- Quando una chiave viene mantenuta costante, lo stato è noto come *typematic poi il rendere il codice* è trasmesso continuamente con una certa velocità. In modalità predefinita la preparazione, una comunicazione tastiera PS2 *fare -Code* circa 100ms dopo il 1 le chiavi sono state mantenute per circa 0,5 s.
- Quando il tasto viene rilasciato, il primo *il codice break (0xF0)* trasmissioni tardi *rendere il codice* di chiavi per identificare quale tasto è stato rilasciato.

#### il trasferimento dei dati Way

Un dispositivo PS2 (tastiera) collegato a KIT DE1 e scambio dati attraverso il percorso dei dati e l'orologio 2



Dati Strada comprende 11 bit



Orologio della via è stato portato in un segnale di clock separata.

I dati saranno trasmessi quando il cambiamento di clock ed è attivamente bassa (fronte di discesa).

### 1.4.3. Monitor VGA

#### **introdurre**

VGA (Video Graphics Array: Video Graphics Array) è stato introdotto da IBM PC basati l'hardware grafico PC e la visualizzazione di progettazione sarà hinh.Chung un'interfaccia consiste di 8 colori base con schermo 640x480 CRT.

#### **meccanismo di base di azione di un CRT**

Schema a blocchi:

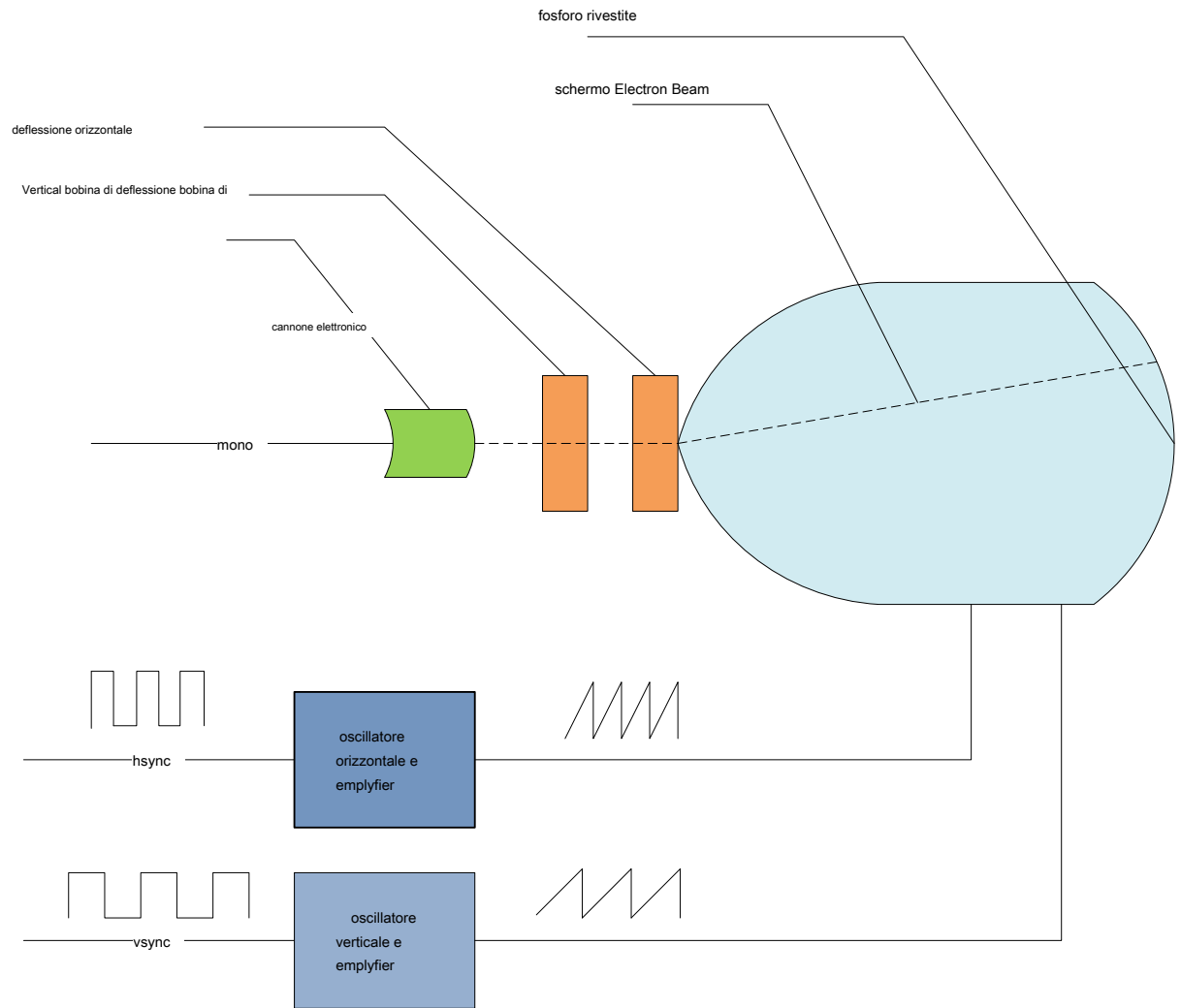
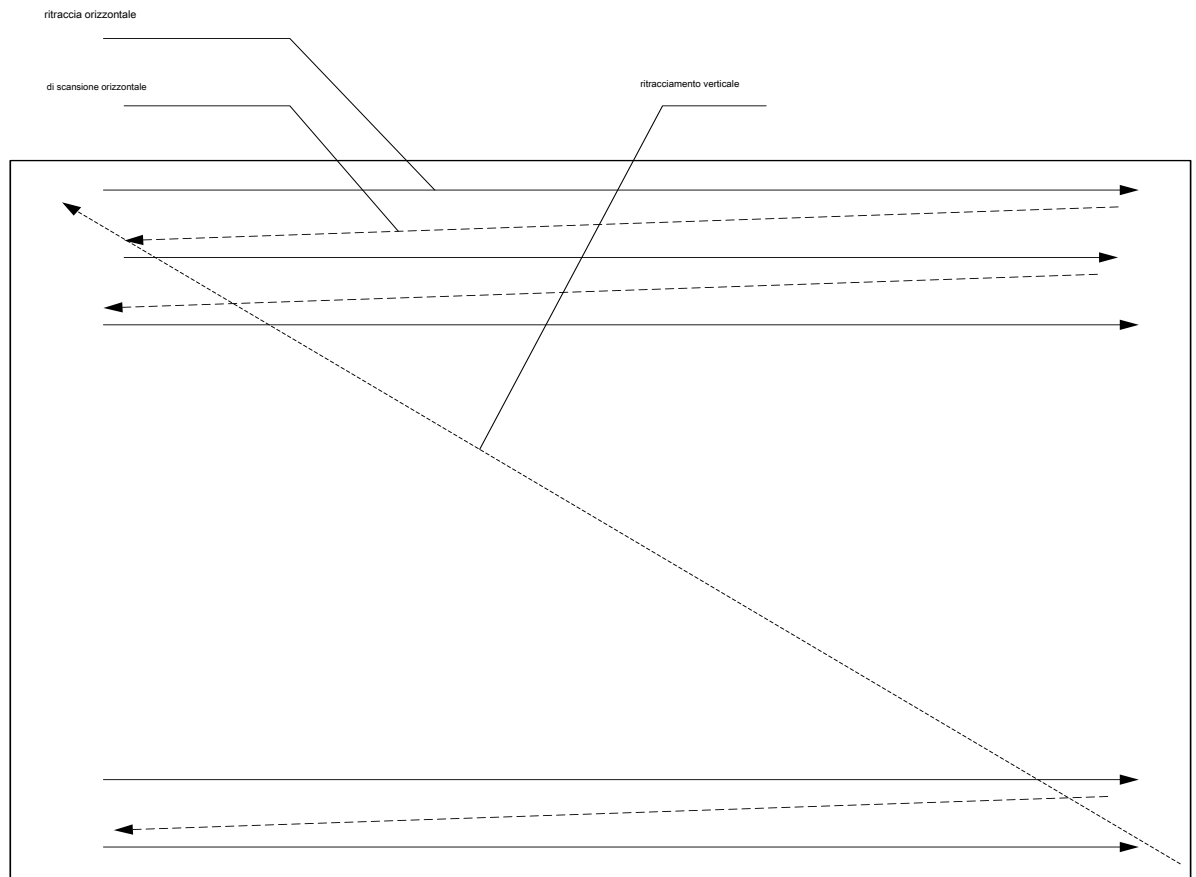


Figura 1

- intensità del fascio di elettroni e la luminosità dello spot è determinato dal livello segnale video di ingresso di tensione, il segnale è il segnale mono. In simile livelli di tensione tra 0 e 0,7 modifiche.
- La bobina di deflessione verticale ed una deflessione orizzontale cruise control coil il flusso di elettroni e fagioli elettroni decidere dove sullo schermo hinh. Voi schermi oggi, gli elettroni con i boss sono guidati da sinistra a destra da cima a fondo.

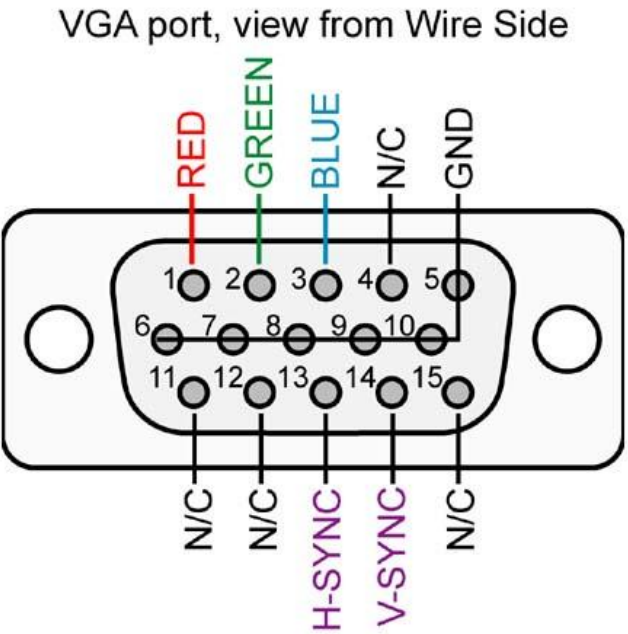
### VGA scansione



Quando la tensione viene applicata alla bobina di deflessione horizontal e aumentare un costantemente, gli elettroni si sposteranno capo dall'angolo sinistra a destra. dopo toccare verso l'angolo destro, i raggi saranno rapidamente coprire sinistro indietro quando la tensione circa 0V (HSYNC). Fino elettroni capo alla stazione a valle controlla la tensione sarà messo nella bobina di deflessione verticale, verranno prese raggi incappucciati torna schermo (VSYNC) e continuare il processo come mostrato in figura

Segnale usato per scansionare la riga schermo hsync e segnale vsync utilizzato per scansione l'intero schermo con frequenza pixel rate 25MHz (25 milioni di pixel realizzazione di 1s) per creare schermo VGA 640x480 risoluzione.

### pin VGA



segnale VGA tra cui 5 attività: due hsync e segnale vsync, tre crediti

segnale video è di colore rosso, blu, verde è collegato a 15 piedi

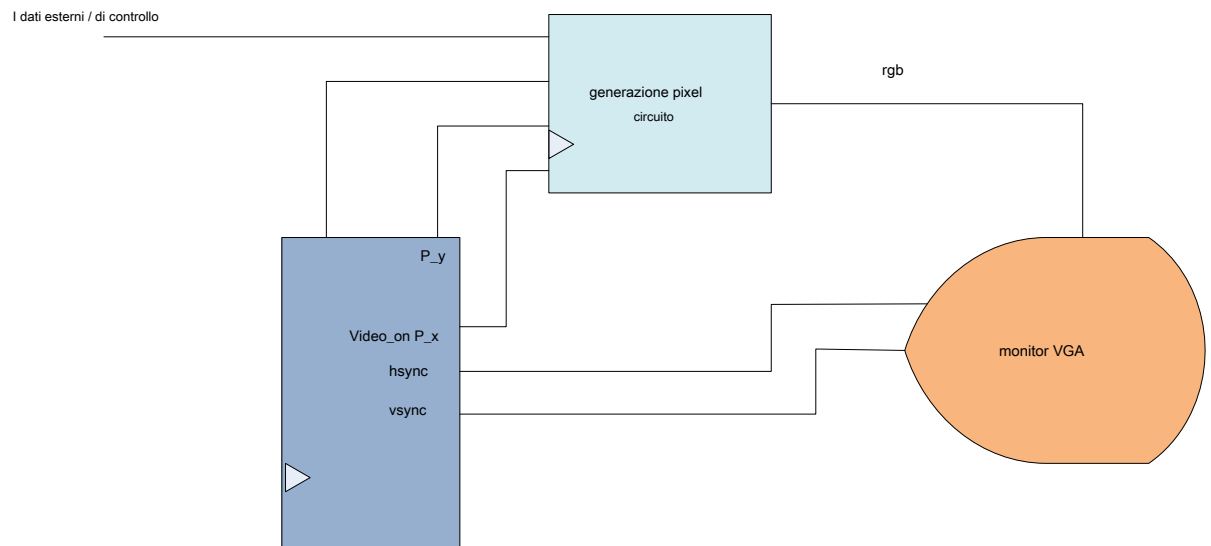
rosso	verde	blu	colore qUA
0	0	0	nero
0	0	1	blu
0	1	0	verde
0	1	1	Ciano
1	0	0	rosso
1	0	1	magenta
1	1	0	giallo
1	1	1	bianco

**il trasferimento dei dati Way**

Per essere in grado di ricevere i dati trasmessi e visualizzati sullo schermo, abbiamo progettato

circuito vga\_sync comprende un timer e il segnale di clock con 2 credito bo.Mach

hsync e segnale vsync è collegato direttamente allo schermo, abbiamo usato per segnali di scansione orizzontale e verticale cane lo schermo hinh. Hai decodificati dalla un contatore disponibile nel segnale di uscita del circuito e 2 è pixel\_x, segnale di uscita pixel\_y. Hai indica il rapporto tra la posizione di scansione e la posizione attuale al punto di un segnale di video\_on anh. Mach al cane o disattivare il display.



### **blocco controllore VGA**

Un circuito per generare il segnale video 3 viene indicato come RGB (rosso verde blu) ingresso è pixel\_x e pixel\_y, video\_on. Il valore di un colore è visualizzato sullo schermo dipende dalla posizione di pixel corrente (pixel\_x e pixel\_y) e dati segnali e cane al di fuori

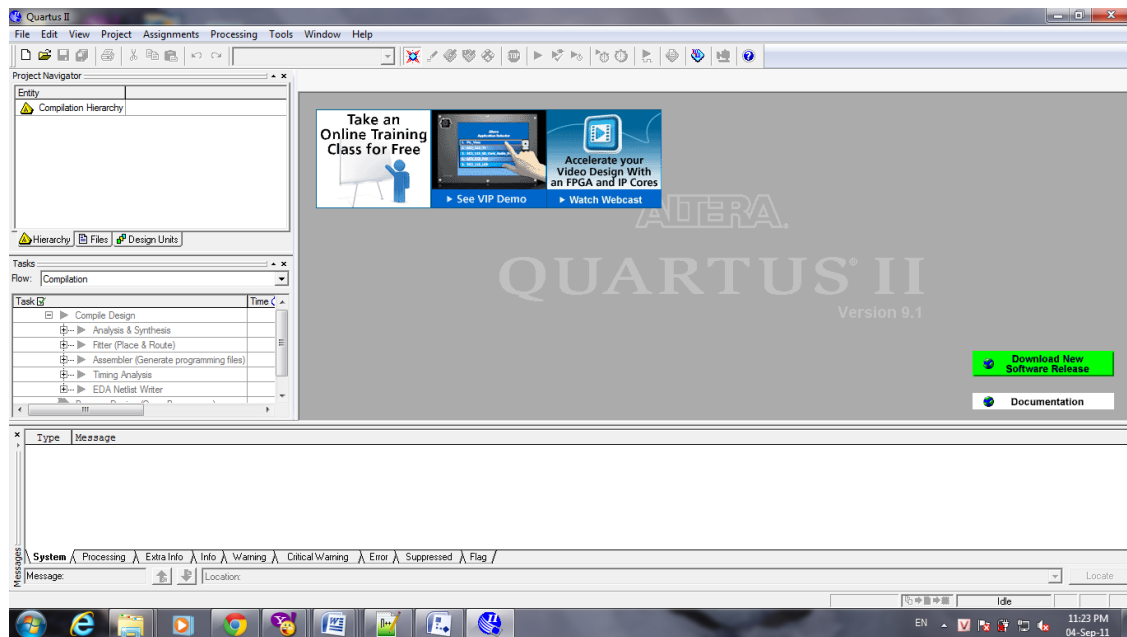
## **1.5. L'introduzione di software**

### **1.5.1. Quartus II**

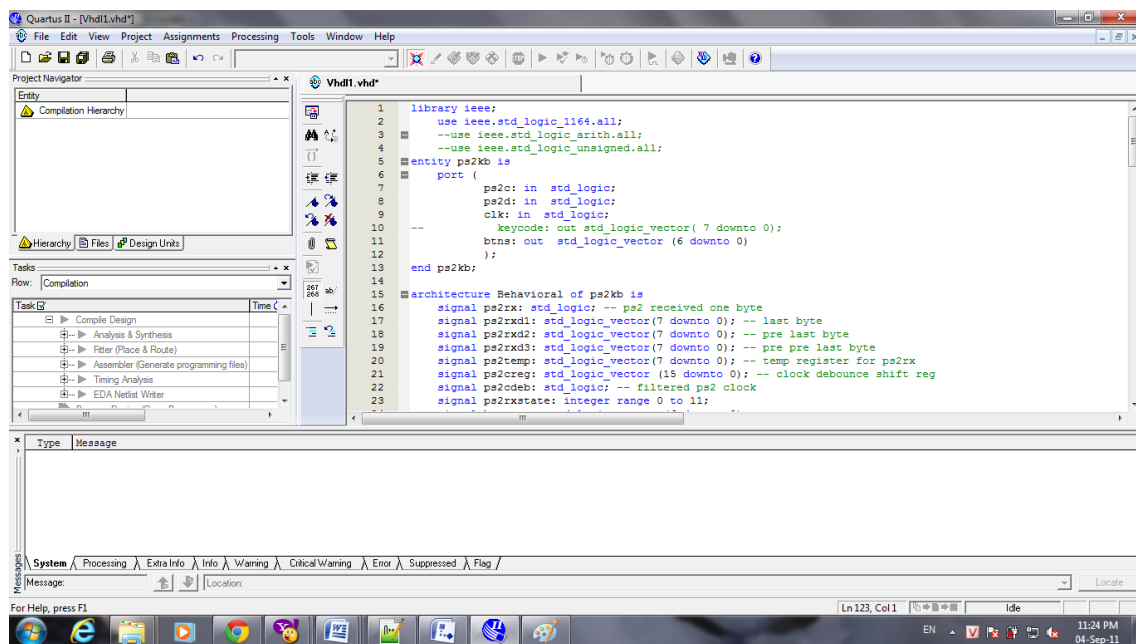
#### **interfaccia principale**



Quartus Web Edition 9.1 per la programmazione e caricati su KIT DE1



### 2.2.1.2 Editor



**assegnare gamba**

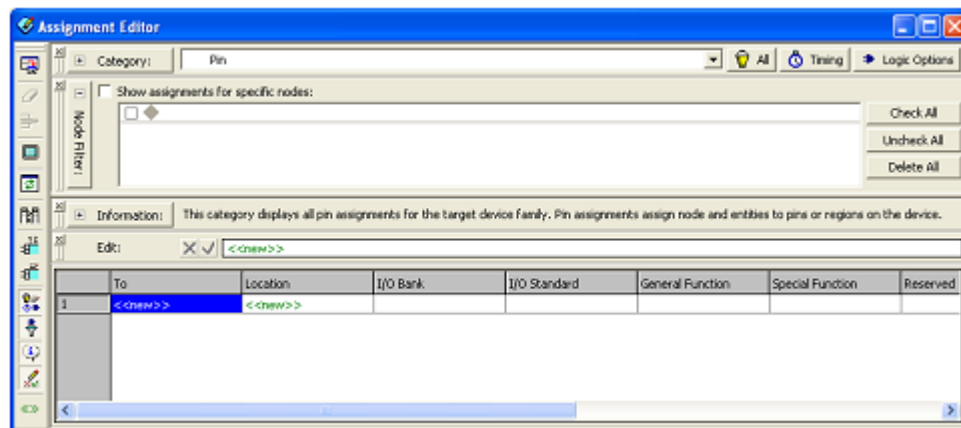
Fase 1: Fare clic su Assegnazioni> Assegnazione Editor. nella categoria

selezionare Pin. Fare doppio clic sul nuovo <>. Clicca segnali dalle liste di diffusione

assegnato a fare la gamba. Quindi, fare doppio clic sulla casella per la casella giusta per

per il segnale da assegnare (colonna Location). Scegliere tra la gamba lista PIN\_XYZ

oppure si può digitare direttamente nella casella Posizione.



**Figura finestra Editor 6. Assegnazione utilizzato per assegnare perni**

Fase 2: Come sopra, perform assegnare alle altre piedini di ingresso

Fase 3: Dopo aver completato i pin assegnati, fare clic su File> Salva. chiuso

finestra dell'editor assegnazione, fare clic su Sì e ricompilare il circuito.

**Nota:** Scegliere un nome che coincide con i segnali della tabella

DE1\_pin\_assignment.csv quando abbiamo solo bisogno di assegnare i pin assignment> import assignment, nel nostro percorso di quel file e premere OK DE1\_pin\_assignment.csv e seguire il punto 3 è xong.Khong prendere tempo per assegnare manualmente sulla mappa.

### **compilare**

Una volta terminato di scrivere codice per un programma che è necessario compilare per creare

I file utilizzati per caricare KIT DE1

Fase 1: Fare clic su Seleziona produzione> Avvia complicazione. compilato in pubblico (o insuccesso) sarà comunicato sul seguente finestra di dialogo unbent quando la compilazione finisce. Confermare facendo clic sul pulsante OK.

Passo 2: Quando la traduzione è completa, viene dato un rapporto compilatore. porta

Questa finestra può anche essere aperto in qualsiasi momento facendo clic scegliere produzione>

Rapporto Complicazione. In questo rapporto include una serie di categorie nel

finestra di sinistra, clicca sulle categorie per vedere i dettagli della

Questa voce compare nella finestra di destra.

Fase 3: correggere i bug

Selezionare Analisi e sintesi> Messaggi per visualizzare i messaggi di errore. fare clic

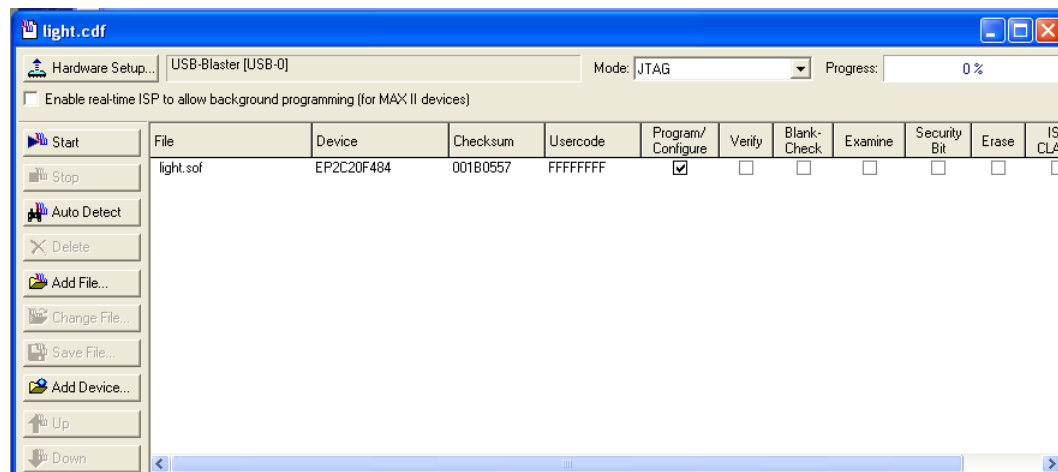
Fare doppio clic sul primo messaggio di errore, gli errori della riga di comando saranno segnati sulla editor di testo, correggerlo e quindi ricompilare il progetto.

## Caricare KIT

Fase 1: spazzolatura commutazione interruttore RUN / PROG a RUN. Fare clic su Strumenti>

Programmer alla finestra come mostrato nella Figura 11. Controllare le opzioni

Programma / configure per consentire xxxxxx.sof file di configurazione caricato.



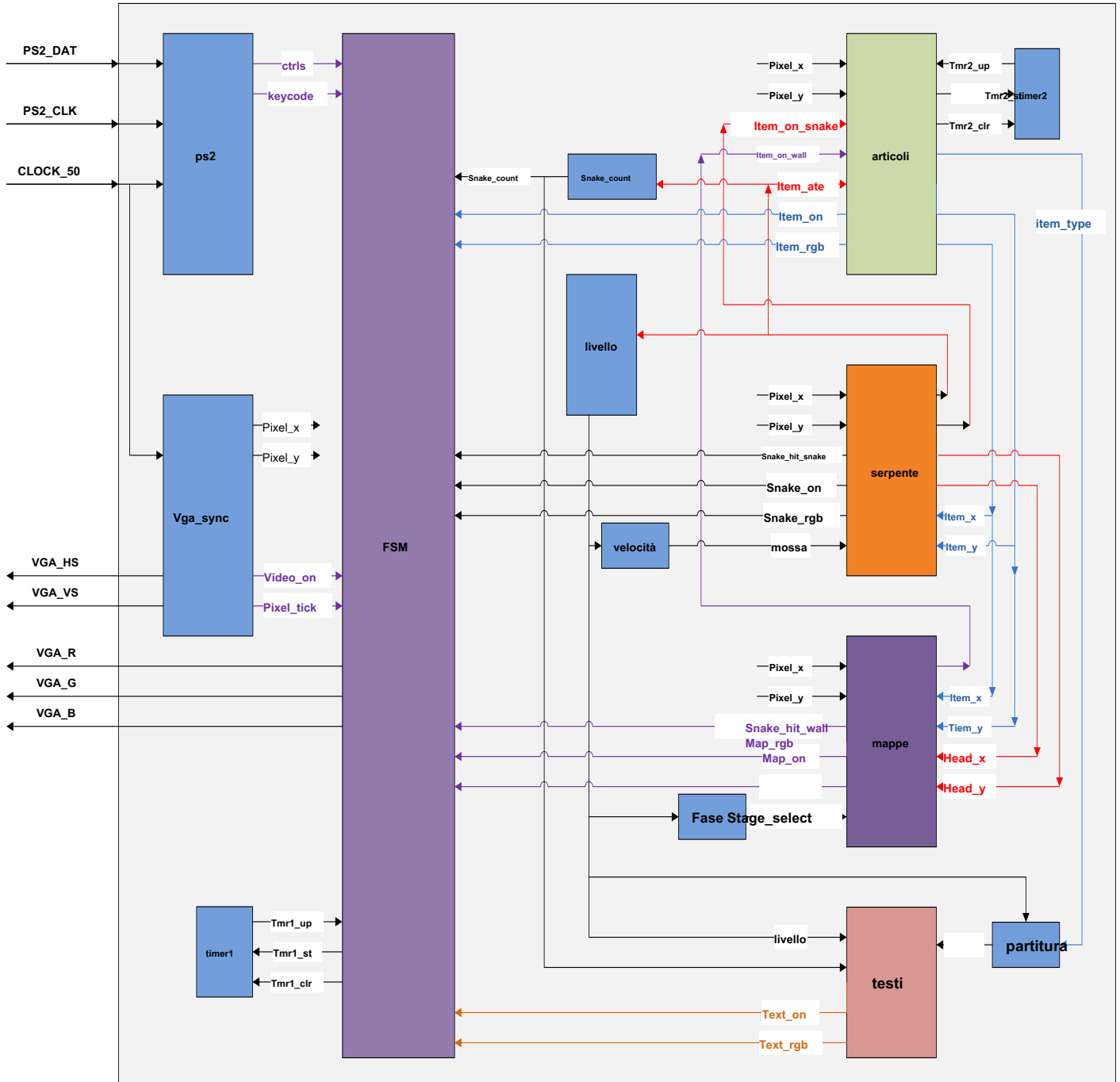
Fase 2: Fare clic sul pulsante Start sulla finestra di sinistra per caricare un file di configurazione in basso

FPGA. Dopo caricato correttamente nella FPGA, controllare questo circuito

implementazione su FPGA per eseguire esattamente come le razze ufficiali del desiderio o meno.

## 1.6. schema a blocchi del sistema

## 1.6.1. sistema ampia



### 1.6.2. Blocchi e funzioni.

unità	funzioni
PS2	Ricezione di segnali dalla tastiera, l'invio di segnali di controllo.
VGA_SYNC	Controllare la visualizzazione sul VGA
ARTICOLI	Creazione di prede per i serpenti.
SNAKE	blocco di controllo del comportamento solido.
MAPPE	La creazione di diversi ostacoli nella schermata di gioco.
testi	Consente di visualizzare le informazioni per l'utente.
FSM	blocco di controllo del sistema, collega ad altri blocchi.
TIMER	Timer, creare latenza.
VELOCITA '	Regolatori velocità del serpente movimento.
LIVELLO	giocatori Calcolo rango.

## 2. particolari di disegno

### 2.1. blocco VGA

#### A proposito di CRT 640x480

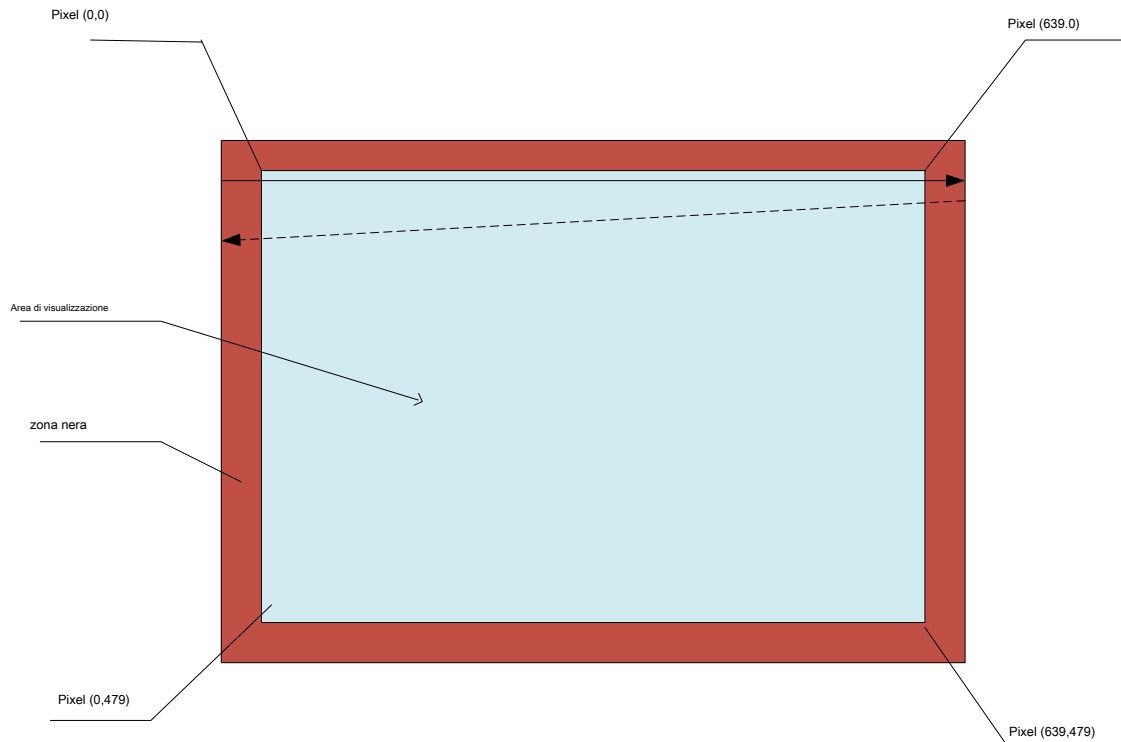
640x480 è il numero di pixel e linee che possono essere visualizzati sullo schermo hinh.Moi

compresi 640 linee orizzontali di pixel, 480 è il numero di linee di scansione orizzontali ma il fatto

sono ciascuno composto da 800 linee orizzontali di pixel e il numero di linee di scansione orizzontali è 525.So

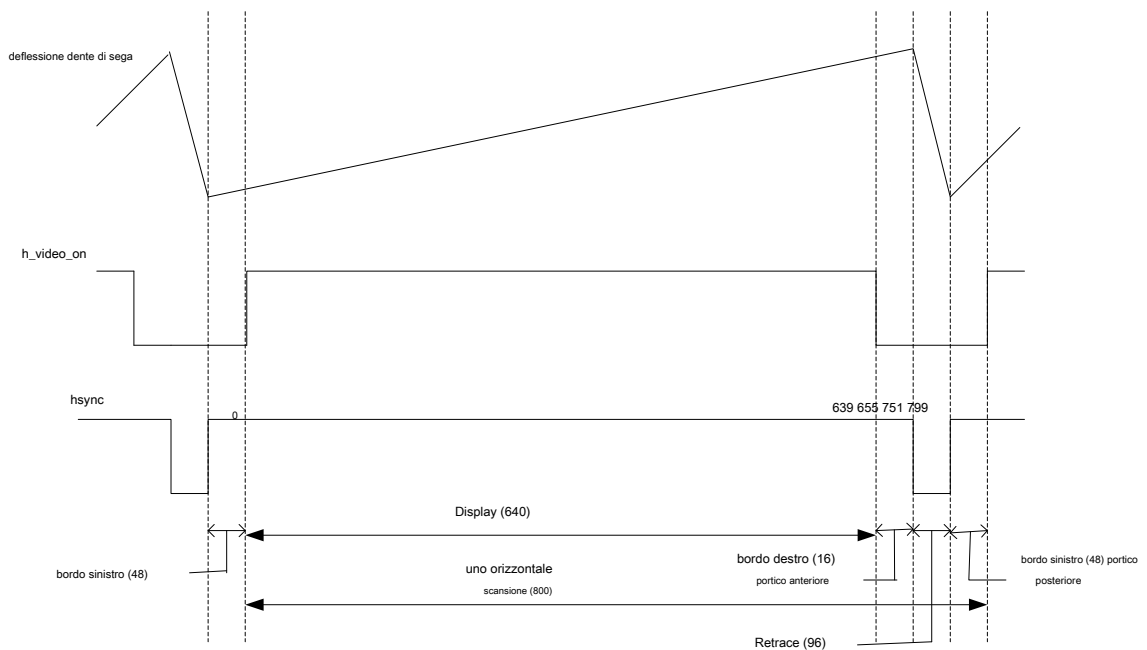
pixel e linee non riportate è chiamato *bordo nero* (bordo nero) .Tan

l'operazione è 25MHz.



### 1.1.1. sincronizzazione asse orizzontale

800 pixel asse orizzontale è diviso in 4 regioni



- Display: regioni in cui i pixel sono indicati con la lunghezza 640px.

- Retrace: le regioni in cui copertina elettrone posteriore sinistro, il segnale video dovrebbe essere off, lunghezza 96px.
- bordo destro (bordo destro) regioni che formano il bordo destro dell'area di visualizzazione è chiamata veranda (cancello), il segnale dovrebbe essere spento, 16px lunghezza
- bordo sinistro (a sinistra): modulo di regione di confine sinistro dell'area di visualizzazione è chiamata portico posteriore (portellone), il segnale deve essere spento, la lunghezza 48px.

Il codice seguente determina le quantità per l'asse orizzontale:

-- 640-by-480 parametri di sincronizzazione VGA

*HD costante: integer: = 640; Area di visualizzazione --horizontal*

*costante HF: integer: = 16; -H. veranda*

*HB costante: integer: = 48; -H. veranda sul retro*

*costante HR: integer: = 96; -H. ripercorrere*

La lunghezza del bordo sinistro e destro può variare fra gli schermi differenti.

Hsync segnale deve circuito più contatore 800 pixel e una risoluzione ma.Khi arrestati prima visualizzare il timer avvia il conteggio e anche segnalare la componente di segnale pixel\_x.Tin hsync segnale basso quando il segnale di uscita del contatore in tra il 656 e il 751.

Abbiamo usato per regolare il segnale video\_on mostra / non mostrare quando il valore di conteggio inferiore a 640

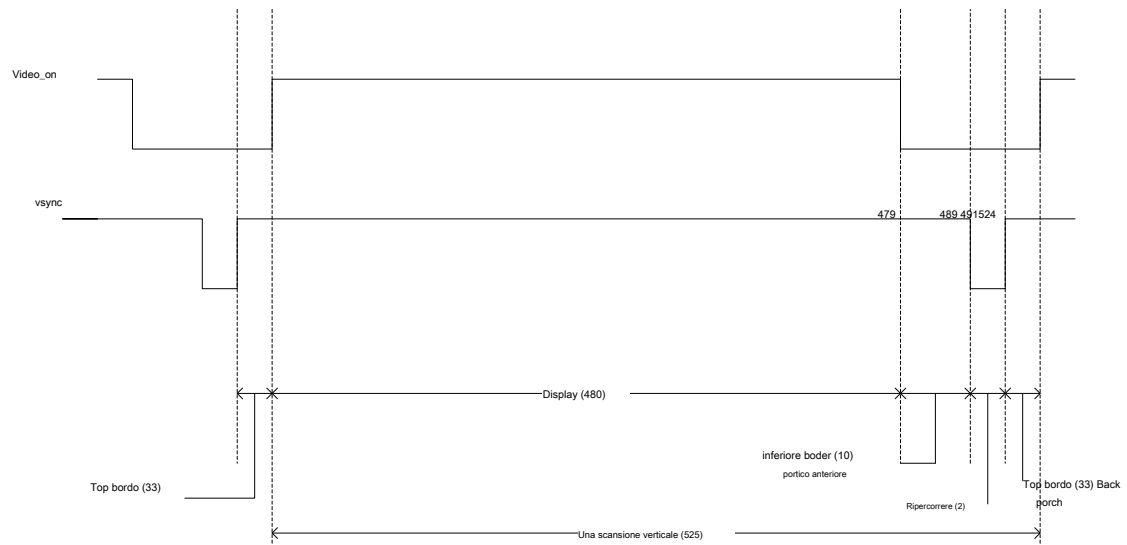
monitor CRT devono essere collocati nel processo di nero e trasparente a destra ea sinistra ripercorrere.

### 1.1.2 sincronizzazione asse verticale

Durante verticale, fascio di elettroni in movimento costante dall'inizio alla fine lo schermo, poi è tornato e tenda di testa richiedono hinh.Su simile un intervallo di rinfresco un'architettura simile vsync hinh.Cau con hsync.

Un ciclo del vsync segnale è 525 linee ed è suddiviso in 4 semi regionale  
come hsync.

Le caratteristiche di ciascuna regione sono hsync simili



linea di segnale Vsync 525 ad un conteggio ed un circuito di conteggio quando il primo ma.Bat

zona partenza thi.tin segnale di corrente del contatore è pixel\_y.Tin segnale vsync in

bassa quando il flusso contatore nelle linee 490 o 491.

Come hsync, usiamo video\_on per vedere / non vedere quando il

valore di conteggio è inferiore a 480.

### 1.1.3 Calcolo del VGA segnali di sincronizzazione di tempo

Nell'ambito di questo progetto, stiamo usando le frequenze che vengono selezionati 25MHz.Su

decisa da tre quantità:

- p: numero di pixel su una linea di scansione orizzontale.  $p = 800 \text{ pixel / linea}$
- l: monitoraggio totale di zucchero.  $l = 525 \text{ linee / schermo}$
- s: numero di fotogrammi al secondo.  $s = 60 \text{ copie / secondo}$



Scegli  $s = 60$  qui è perché l'occhio umano funziona bene in questa cornice e contro il lampeggiante.

Così pixel rate =  $p * l * s = 25\text{M}$  (pixel / secondo)

#### 1.1.4 Completa VGA grafica

Sopra abbiamo progettato 2 set dem. Un problema di progettazione qui è che solo il supporto KIT DE1

Supporto di frequenza 50MHz tale richiesta è 25MHz. Vi dai requisiti di progettazione in modo da creare

1 di 25MHz consentendo al marcatore per disattivare o attivare il dem. Tin

segnale p\_tick è un segnale per eseguire questo lavoro e coordinare le attività

circuito di generazione pixel.

Utilizzare 2 reale avanzamento h\_end e v\_end controllare il completamento della scansione orizzontale e verticale.

Anche per evitare interferenze situazione abbiamo bisogno di utilizzare più del

Tampone è stato inserito segnali HSYNC e VSYNC.

Il seguente codice utilizzato per creare il contatore mod-2 usato per segnare il passo:

```
-- circuito mod-2 per generare 25 MHz consentono tick
```

```
mod2_next <= non mod2_reg;
```

```
-- 25 MHz di pixel tick
```

```
pixel_tick <= '1' quando mod2_reg = '1' altro '0';
```

Il seguente codice utilizzato per identificare il completamento del orizzontale:

```
h_end <= - fine del contatore orizzontale
```

```
'1' quando h_count_reg = (HD + HF + HB + HR-1) --799 altro '0'; // sottrarre 1 perché
```

```
contiamo da 0 //
```

Il codice seguente viene utilizzato per eliminare il rumore:

```
-- sincronizzazione orizzontale e verticale, tamponata per evitare inconveniente
```

```
h_sync_next <= '1', quando (h_count_reg >= (HD + HF)) -- 656
```

```
e (h_count_reg <= (HD + HF + HR-1)) else -751
```

'0';

## 2.2. blocco PS2

Sul circuito comprende un ricevitore che può inserire un blocco contro interferenza del segnale, un blocco FSM al cane codice di scansione trasmissione, blocchi FIFO e trasmettere i dati a bruciare sotto la first in first out (first in first out).

### 2.1 Blocco ricezione dei dati (PS2\_rx)

Utilizzare il seguente codice per evitare interferenze con il segnale:

```
ps2cdeb <= '0' quando ps2creg = "1111111111111111" else // restituisce 0 se tutto
bit è 1
```

```
'1' quando ps2creg = "0000000000000000" else // restituisce 1 quando tutti i bit sono 0;
```

Per capire il meccanismo della PS2 ricevere dati consideriamo il seguente diagramma di flusso:

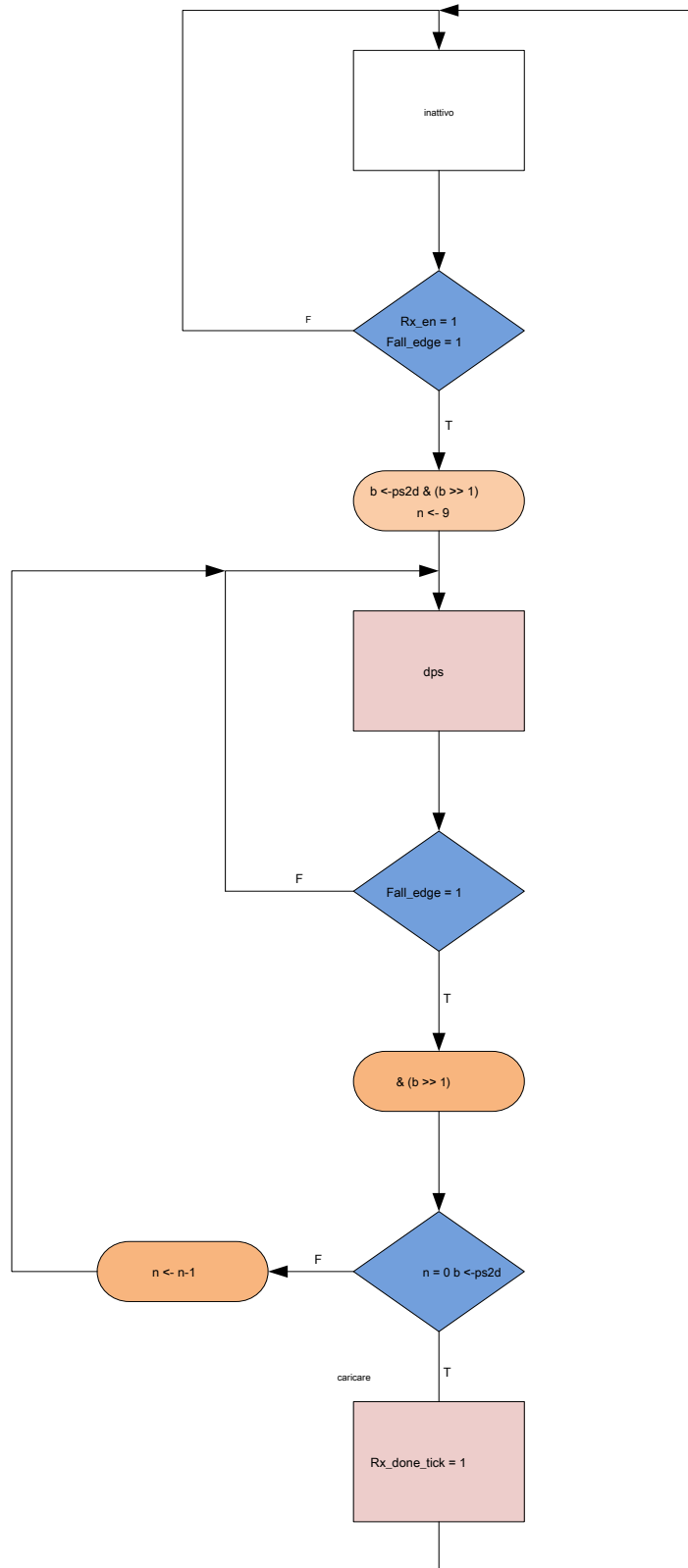


grafico ASMD del ricevitore porta PS2

Quando il segnale ricevuto consente segnali positivi rx\_en e costolette giù stato

Stato si tradurrà per iniziare po 'e passare a dichiarare DPS.

Poiché i dati vengono ricevuti in un blocco è stato controllato, tradurremo

10bit di nuovo in uno status separato piuttosto che utilizzare uno stato separato

come dati, parità, stop.

Poi andrà al carico circuito stato in cui un ciclo di clock

distribuito più per completare il processo di traduzione del bit di stop, e il segnale

rx\_done\_tick essere inserito dopo il primo ciclo a notare di aver ricevuto i dati

Dati.

### 2.3. ARTICOLI blocco

#### 2.3.1. Descrizione generale

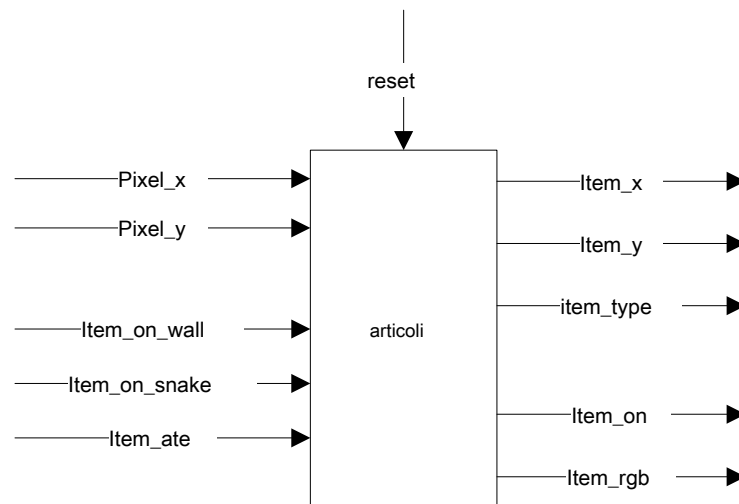


Figura 2. ARTICOLI blocchi

ARTICOLI incaricati blocchi nato preda di serpenti, la nuova posizione incontra il condizioni:

- Nuova esca viene creata se l'esca è mangiata, o non appena sono stati creati

Il nuovo bug un certo combustibili solidi, o centrato intorno alle ostruzioni.

- Bait è generato in posizioni casuali, non dipende dalla posizione di fondo in precedenza.
- Nuovo Predator nato nella regione devono spostare i serpenti.
- Nuovo primer è la posizione non coincide con qualsiasi combustibile solido, se solida comprensione su, per ricreare una nuova esca.
- Nuovi ostacoli esca nati non devono essere identici, se non, per creare la nuova preda.
- Le esche speciali scompaiono dopo un certo periodo di tempo regolamenti.

Tipo	Forma	Tempo	Caratteristiche	Le donne scomparse	
1	mele			1	
2	come	5s		1	decelerazione
3	cuore	5s		1	Aumento giochi di nuovo
--	--	--	--	--	--

Gli ingressi e le uscite sono dichiarati come segue:

```

I prodotti che entità è
porte (
-- orologio principale
clk: in std_logic;
reset: in std_logic;
-- coordinate del pixel sottoposti a scansione
pixel_x: in std_logic_vector (9 downto 0);
pixel_y: in std_logic_vector (9 downto 0);
-- attuale livello di apparire diverso livello adeguato fondo

```

```

        Livello: nel campo dei numeri interi da 1 a 7;

        item_x: fuori senza segno (9 downto 0);

        item_y: fuori senza segno (9 downto 0);

        pausa: in std_logic;

        item_on_wall: in std_logic;

        item_on_snake: in std_logic;

        item_ate: in std_logic;

        timer2_reset: out std_logic;

        timer2_start: out std_logic;

        timer2_up: in std_logic;

        item_type: fuori campo dei numeri interi da 1 a 7: = 1;

        item_on: out std_logic;

        item_rgb: out std_logic_vector (2 downto 0)

    );

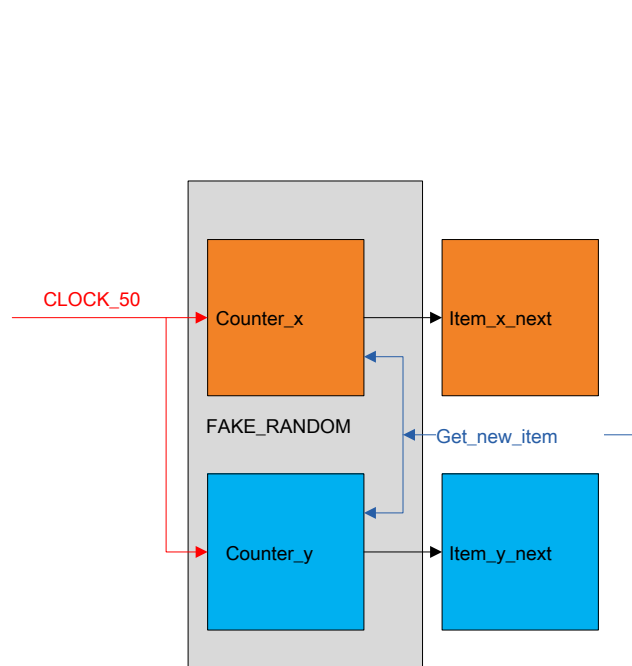
```

articoli finiti;

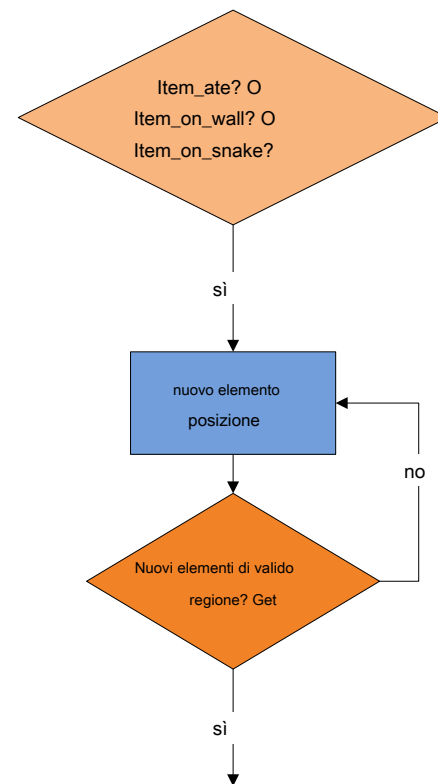
Creare rom articoli:

Descriviamo ROM attraverso la matrice di bit, come segue:

Generatore di numeri casuali:



Usiamo due contatori come generatore di numeri casuali. con orologio 50Mhz, e gli elementi di tempo sconosciuto mangiato khi, valore del contatore sembrano essere casuale.



Articolo controllato la posizione fino Phai  
ABBIAMO un valore valido

Per generare numeri casuali usiamo i valori di rotazione e contatore contatore  
essere recuperate in qualsiasi momento che vogliamo, per garantire la casualità, è  
Contatori tempo di consegna è molto più piccolo rispetto al tempo medio  
ottenere il valore dal contatore  
Io uso il contatore con un valore piccolo, ma ripetono con frequenza in fretta, poi  
è l'orologio 50Mhz.  
Contatore 1 viene utilizzato per la gamma di valori in base alle coordinate x  
cross-priming  
Contatori 2 usati per creare coordinata y valori verticalmente  
di prede  
Combina 2 valori stanno per ottenere le coordinate della nuova esca è nato,  
e sembra quasi casuale.

codice:

## 2.4. blocco SNAKE

### 2.4.1. diagrammi

### 2.4.2. codice

## 2.5. blocco MAPPE

## 2.6. testi di blocco

### 2.6.1. font

- ROM con lettura synchronous (inferire Block RAM)
- carattere ROM
  - - 8-by-16 caratteri ( $8\text{-by-}2^4$ )
  - - 128 ( $2^7$ ) caratteri
  - - dimensioni ROM: 512-by-8 ( $2^{11} \times 8$ ) bit
- 16K bit: 1 BRAM

IEEE biblioteca;

ieee.std\_logic\_1164.all utilizzare;

ieee.numeric\_std.all utilizzare;

entità font\_rom è

porte (

clk: in std\_logic;

addr: in std\_logic\_vector (10 downto 0);



Dati: out std\_logic\_vector (0-7)

);

font\_rom fine;

architettura arco di font\_rom è

ADDR\_WIDTH costante: integer: = 11;

DATA\_WIDTH costante: integer: = 8;

segnale addr\_reg: std\_logic\_vector (1 downto ADDR\_WIDTH-0);

Tipo rom\_type è array (0 e 2 \*\* ADDR\_WIDTH-1)

di std\_logic\_vector (0 a DATA\_WIDTH-1);

-- definizione ROM

costante ROM: rom\_type: = (- 2 ^ 11-by-8

"00000000", - 0

"00000000" - 1

"00000000" - 2

"00000000", - 3

"00000000", - 4

"00000000" - 5

"00000000" - 6

"00000000" - 7

"00000000", - 8

"00000000", - 9

"00000000" - un

"00000000", - b

"00000000", - c

"00000000", - d

```

"00000000", - e
"00000000" - f

-- codice x01

"00000000", - 0
"00000000" - 1
"01111110" - 2 *****
"10000001", - 3 *   *
"10100101", - 4 * * * *
"10000001", - 5 *   *
"10000001" - 6 *   *
"10111101" - 7 * * * * *
"10011001", - 8 * * * *
"10000001" - 9 *   *
"10000001" - a *   *
"01111110", - b *****
"00000000", - c
"00000000", - d
"00000000", - e
"00000000" - f

...

);

iniziare

-- addr registrati per inferire blocco RAM

processo (CLK)

iniziare

se (clk'event e clk = '1') allora

```

```

        addr_reg <= addr;

    end if;

    fine processo;

    Dati <= ROM (to_integer (senza segno (addr_reg)));

end arco;

```

## 2.7. blocco FSM

### 2.7.1. diagrammi

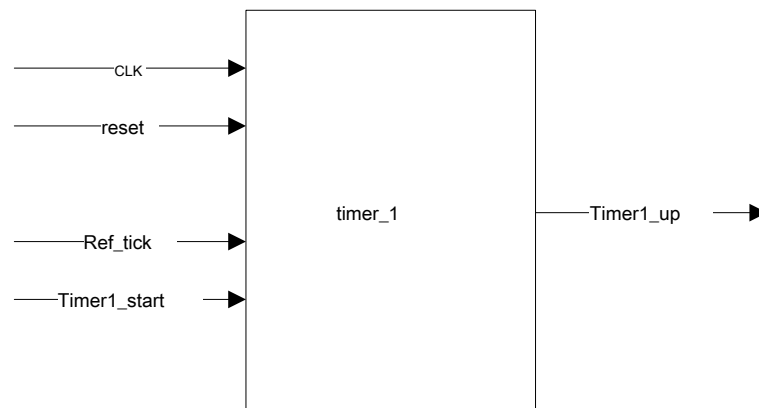
### 2.7.2. codice

## 2.8. sottoblocco

### 2.8.1. timer\_1

Si tratta di blocchi 2s timer per la creazione di ritardo tra lo stato della macchina stati FSM, utile quando si desidera notare qualcosa sullo schermo. in Ecco l'annuncio GAME OVER, STAGE CLEAR LEVEL UP ...

Modello:



Dichiarare ingresso, di uscita:

Timer entità è

```

porte (

    -- Main-orologio in ingresso 50MHz

    CLK:                in std_logic;

```

```

-- reset sincrono

reset:          in std_logic;

-- tasso di 60Hz-clock del monitor è di input

refr_tick:      in std_logic;

-- segnale di partenza

timer_start: in std_logic;

-- uscita

timer_up:       std_logic out

);

end timer;

```

Useremo il contatore per fare missione a tempo, così abbiamo salvato

il valore di conteggio in un registro, può contare salita, ma come arbitraria riduzione conteggio  
riduzione del numero è più vantaggioso quando si confrontano il valore del registro con il valore 0.

Qui abbiamo bisogno di circa due secondi il timer, immettere un segnale  
aggiornare la schermata a circa 60Hz a 2 secondi corrisponderebbe ai valori  
dei registri 120.

Abbiamo scelto registri 7-bit, il che significa il suo valore massimo è 127, ossia il periodo  
 $127/60 = 2.1$  lag è secondi, incertezza ancora accettabile.

codice:

architettura arco del timer è

```

-- valore massimo nel registro è 127, il valore massimo è circa 2 secondi contatore

segnale timer_reg, timer_next: senza segno (6 downto 0);

iniziare

```

La sezione di codice tempo di lavorazione successivo conteggio, primi registri del processore, con l'orologio 50Mhz come orologio principale per aumentare il sistema di sincronizzazione sistema

Ci schieriamo contatore formato macchina dello Stato, ma senza alcun creato Stato registro RAM unica cosa che vale la pena diviso in due bar

annotazione: timer\_reg solo valore corrente del contatore, e il valore del contatore unico timer\_next poi, se vi capita di clock.

codice:

```
-- registri

processo (clk, reset)

iniziare

    se (clk'event e clk = '1') allora

        Se il reset = chiave '1'

            timer_reg <= (altri => '1');

        altro

            timer_reg <= timer_next;

        end if;

    end if;

fine processo;

-- logica dello stato successivo

processo (timer_start, timer_reg, refr_tick)

iniziare

    if (timer_start = '1') allora

        timer_next <= (altri => '1');

    elsif refr_tick = '1' e timer_reg /= 0 allora

        timer_next <= timer_reg - 1;
```

```

        altro

        timer_next <= timer_reg;

    end if;

    fine processo;

    -- produzione

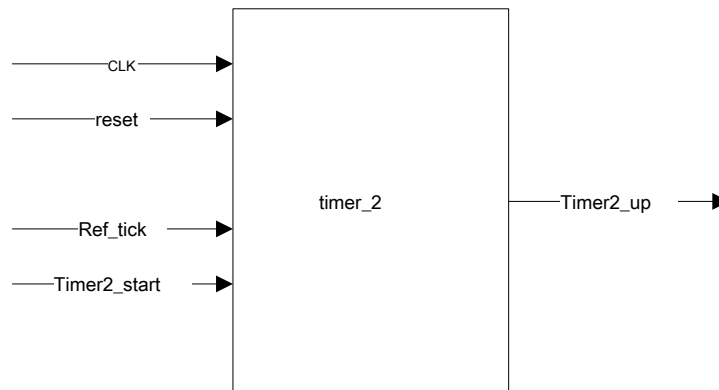
    timer_up <= '1' quando timer_reg = 0 altrimenti '0';

end arco;

```

### 2.8.2. timer\_2

Analogamente timer\_1, timer\_2 anche il contatore, ma è utilizzato in altri scopi, è esposizione di tempo di un particolare tipo di esca nel gioco, come ad LIVE\_UP, SPEED\_DOWN ... questa nuova specie visualizzato in un certo periodo di tempo e poi scomparire se durante che il tempo non riusciva a mangiare cibi solidi.



Come timer\_1, contiamo l'ultima volta con il registro orologio battito 60Hz attraverso il valore del registro. L'unica differenza è la lunghezza della barra valore massimo ovvero rilevate che registri di memorizzazione.

Qui registra sarà:

```
segnale timer2_reg, timer2_next: senza segno (9 downto 0);
```

10 valore di bit memorizzati max = 511, quindi il più grande ritardo è  $511/60 = 9$  secondi.

La movimentazione timer\_2 timer\_1 identico non deve dire qui nuovamente.

### 2.8.3. LIVELLO

A livello di blocco monitorare il numero di esche solido è stato mangiato, in modo che il cibo quando solidi una certa quantità di esche, il rango del giocatore saranno aumentate, accompagnati per cui la velocità di movimento più mirato aumento solido il livello di difficoltà per lo schermo giocare.

Una volta che i giocatori iniziare a livello 1 e raggiungere il livello 5, il gioco a schermo cambierà, abbiamo bisogno di controllare quando cambia schermo di gioco, poi Questo dovrebbe aggiungere un po 'di codice per un esame del livello raggiunto essere.

I pin di segnale necessari sono descritti come segue:

Entità è

```

porte (
    -- orologio principale
    clk: in std_logic;

    riavviare: in std_logic_vector (1 downto 0);

    -- esca è stato mangiato
    item_ate: in std_logic;

    -- segnale di notifica di livello
    level_up: out std_logic;
```

-- segnale sugli schermi.

stage\_clear: out std\_logic;

-- schermata di gioco attuale del lettore.

stage\_select\_in: nel campo dei numeri interi da 1 a 7;

stage\_select\_out: fuori campo dei numeri interi da 1 a 7;

-- informare livello attuale del giocatore.

Livello: fuori campo dei numeri interi da 1 a 7

);

end livello;

Codice a livello di blocco per controllare il lettore sulla base del esca è stato mangiato.

Alcuni esca per mangiare il livello successivo è 5 volte superiore rispetto ai livelli attuali, per esempio dal livello del

Livello 2, livello 3 dovrebbe desiderare di mangiare  $2 * 5 = 10$  primer.

elsif item\_ate = chiave '1'

se counter\_next\_level =  $5 * \text{level\_tmp}$  Athens

level\_tmp: = level\_tmp + 1;

Questo codice controllerà la schermata di gioco se il giocatore ha raggiunto il livello 6, il

I giocatori saranno sollevate sceneggiatura più difficile

se level\_tmp = 6 Athens

se stage\_sel\_tmp < 2 bar

stage\_sel\_tmp: = stage\_sel\_tmp + 1;

end if;

stage\_clear\_tmp: = '1';

e gradi contenute nella nuova schermata di gioco sarà avviato di nuovo dal livello 1.

level\_tmp: = 1;

end if;



```

        level_up_tmp: = '1';

        counter_next_level: = 0;

        altro

        counter_next_level: = counter_next_level + 1;

        end if;

    end if;

    Livello <= level_tmp;

    level_up <= level_up_tmp;

    stage_clear <= stage_clear_tmp;

    stage_select_out <= stage_sel_tmp;

end if;

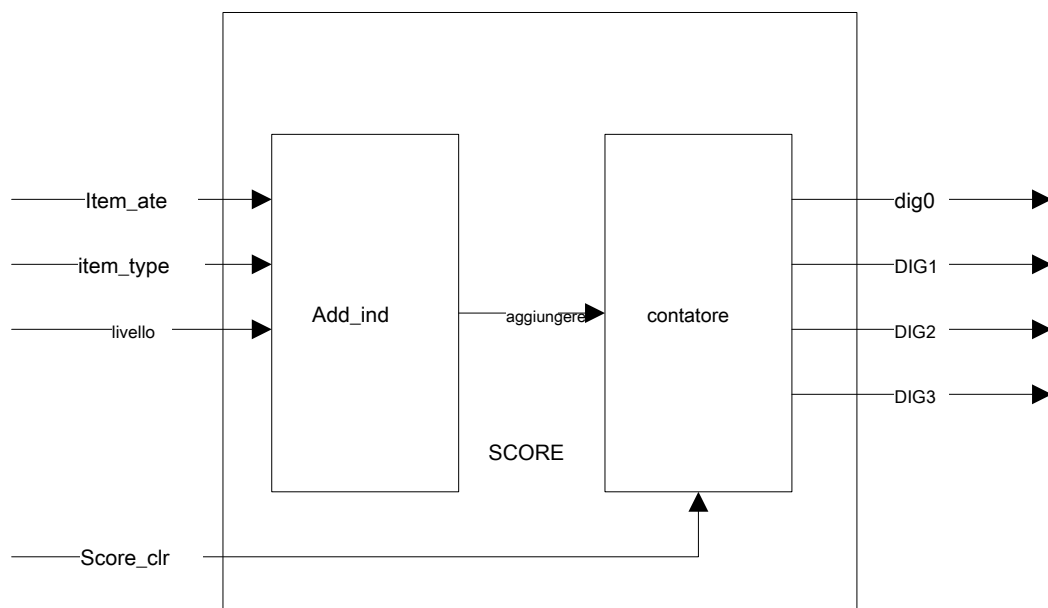
fine processo;

end arco;

```

## 2.8.4. PUNTEGGI

### 2.8.4.1. diagrammi



- Questa unità è responsabile per la classificazione del giocatore, e il punteggio ricevuto dipenderà dal tipo di esca che serpenti mangiano, il livello corrente del solido (velocità il più veloce, più alto è il livello, più punti ricevuti capiscono).
- Blocchi ricevono segnali da parte degli investitori compresi gli oggetti Item\_ate, e item\_type, dopo quando si calcola il gioco blocco Add\_ind più punto che nguii inoltre, si creerà il componente aggiuntivo per bloccare punti di impulso del contatore. Per esempio quando Item\_ate = 1, item\_type = 3, Livello = 2, il punteggio è di 3 + 2 persone = 5, questa volta genererà 5 segnale Add\_Ind cubic Aggiungi al contatore.
- Score\_lcr rivolge ripristinare quando il punteggio della schermata di gioco appena catturato 0000 testa

#### 2.8.4.2. Counter\_9999

Come menzionato sopra, più segnali di blocco ricevuti attraverso COUNTER\_9999 piede d\_inc, ogni volta che questo segnale da un basso ad un valore alto del contatore aumenterà di 1.

Le gambe del blocco sono descritti come segue:

entità counter9999 è

```

porte (
    -- orologio principale
    clk: in std_logic;

    -- Reset
    reset: in std_logic;

    -- Inoltre indicano
    d_inc: in std_logic;

    -- uscita è il numero di vantaggi per l'inserimento di

```

Testi di trasformarsi in cifre, serve la visualizzazione sullo schermo.

```

        dig0: out std_logic_vector (3 downto 0);

        DIG1: out std_logic_vector (3 downto 0);

        DIG2: out std_logic_vector (3 downto 0);

        DIG3: out std_logic_vector (3 downto 0)

    );

end counter9999;

-----

architettura arco di counter9999 è

    -- registri salvare il valore corrente e il successivo conteggio 4.

    segnale dig0_reg, dig1_reg, dig2_reg, dig3_reg: senza segno (3 downto 0);

    segnale dig0_next, dig1_next, dig2_next, dig3_next: unsigned (3 downto
0);

iniziare

-- registri

    processo (clk, reset)

        iniziare

            se (clk'event e clk = '1') allora

                Se il reset = chiave '1'

                    dig0_reg <= (altri => '0');

                    dig1_reg <= (altri => '0');

                    dig2_reg <= (altri => '0');

                    dig3_reg <= (altri => '0');

                altro

                    dig0_reg <= dig0_next;

                    dig1_reg <= dig1_next;

                    dig2_reg <= dig2_next;

```

```
        dig3_reg <= dig3_next;
    end if;

end if;

fine processo;

-- logica dello stato successivo per il contatore decimale
processo (reset, d_inc, dig0_reg, dig1_reg, dig2_reg, dig3_reg)
    iniziare

    Se il reset = chiave '1'

        dig0_next <= (altri => '0');
        dig1_next <= (altri => '0');
        dig2_next <= (altri => '0');
        dig3_next <= (altri => '0');

    altro

        dig0_next <= dig0_reg;
        dig1_next <= dig1_reg;
        dig2_next <= dig2_reg;
        dig3_next <= dig3_reg;

    end if;

    se (d_inc = '1') allora

        se dig0_reg = 9 bar

            dig0_next <= (altri => '0');

            se dig1_reg = 9 bar

                dig1_next <= (altri => '0');

                se dig2_reg = 9 bar

                    dig2_next <= (altri => '0');
```

```

        se dig3_reg = 9 bar
            dig3_next <= (altri => '0');
        altro
            dig3_next <= dig3_reg + 1;
        end if;

    altro
        dig2_next <= dig2_reg + 1;
    end if;

    altro
        dig1_next <= dig1_reg + 1;
    end if;

    altro
        dig0_next <= dig0_reg + 1;
    end if;

end if;

fine processo;

-- produzione
dig0 <= std_logic_vector (dig0_reg);
DIG1 <= std_logic_vector (dig1_reg);
DIG2 <= std_logic_vector (dig2_reg);
DIG3 <= std_logic_vector (dig3_reg);

end arco;

```

#### 2.8.4.3. unità punteggi

Questa massa è la natura massa del blocco di controllo e counter999

comunicare con il blocco superiore.

Dichiarare i pin di segnale come segue:

```

IEEE biblioteca;

ieee.std_logic_1164.all utilizzare;

punteggio entità è

porte (

    -- orologio principale

    clk: in std_logic;

    -- punto di segnale cancellato del 0000

    score_clr: in std_logic;

    -- Giornali esca è stata mangiata

    item_ate: in std_logic;

    -- esche

    item_type: nel campo dei numeri interi da 1 a 7;

    -- Gli attuali livelli di giocatori

    Livello: nel campo dei numeri interi da 1 a 7;

    -- messo fuori per la cifra testi

    dig0: out std_logic_vector (3 downto 0);

    DIG1: out std_logic_vector (3 downto 0);

    DIG2: out std_logic_vector (3 downto 0);

    DIG3: out std_logic_vector (3 downto 0)

);

punteggio finale;

```

dichiariamo un componente per il blocco **counter9999**

ed eseguire segnale morsettiera.

```
counter_u0: work.counter9999 Entity
```

```
Mappa porto (...);
```

Parte importante qui è come creare ritmo urbano corrispondente numero di punti

saranno aggiunti al giocatore, questo tasso di conteggio tiene

**counter9999** tramite contatore indicatori di direzione e d\_inc.

Abbiamo gestito attraverso un processo come segue:

```
processo (clk, livello, item_ate, item_type)
```

```
-- condizioni di flusso variabile
```

```
aggiungere variabili: std_logic;
```

```
-- valore memorizzato alla comunità
```

```
variabile contatore: campo dei numeri interi da 0 a 63;
```

```
iniziare
```

```
se clk'event e clk = chiave '1'
```

```
-- se l'esca è mangiato
```

```
se item_ate = chiave '1'
```

```
-- uscita del contatore segnale
```

```
aggiungere: = '1';
```

```
end if;
```

```
-- se il segnale per il contatore
```

```
se add key = '1'
```

```
-- calcolare la quantità più
```

```
se il contatore = livello + item_type Athens
```

```
-- Se un numero sufficiente scrittori giornale della comunità
```

```
comunità
```

```
aggiungere: = '0';
```

```
-- mettere contatore a 0
```

```
contatore: = 0;
```

```
altro
```

```
-- in caso contrario, il conteggio successivo  
contatore: = counter + 1;  
  
end if;  
  
end if;  
  
-- e fornire segnali per counter9999  
d_inc <= aggiungere;  
  
end if;  
  
fine processo;  
  
end arco;
```

### 2.8.5. costi

### 2.8.6. SNAKE\_COUNTER