

Realizzare sistemi di sorveglianza HD con FPGA a basso costo

Suhel Dhanani
Senior manager - DSP
Altera

Mankit Lo
Ceo
EyeLytics

Gli FPGA delle più recenti generazioni permettono di realizzare telecamere ad alte prestazioni ottimizzate in termini sia di consumi sia di costi

Quello della sorveglianza è il più recente segmento del mercato video in cui ha iniziato ad affermarsi e diffondersi il concetto di alta definizione (HD) che, ovviamente, richiede capacità di elaborazione dei segnali video ad alte prestazioni. Le telecamere di sicurezza HD devono codificare direttamente l'immagine mediante un codificatore H.264 (uno standard di compressione video noto anche come MPEG-4 AVC - Advanced Video Coding) di elevata qualità in modo da rendere possibile la trasmissione di un'immagine realistica su un collegamento Ethernet standard.

Un encoder H.264 Main Profile che può codificare un flusso video HD in tempo reale richiede una piattaforma su silicio che, a fronte di una struttura di elaborazione del segnale ad alte prestazioni, assicuri costi e consumi contenuti. Per soddisfare tali richieste gli FPGA a basso costo delle più recenti generazioni rappresentano senza dubbio la soluzione ideale.

Nel corso dell'articolo viene descritta l'architettura di una telecamera di sor-

veglianza ad alta definizione e la sua realizzazione pratica mediante un FPGA low-cost.

Architettura della telecamera HD

Le più recenti telecamere IP ad alta definizione sono i punti terminali di un sistema digitale connesso in una rete IP. Queste telecamere acquisiscono i video ad alta definizione, eseguono le operazioni di pre-elaborazione e codifica e inviano il flusso codificato mediante una connessione Ethernet. Tutte le funzioni di elaborazione del segnale devono esse-

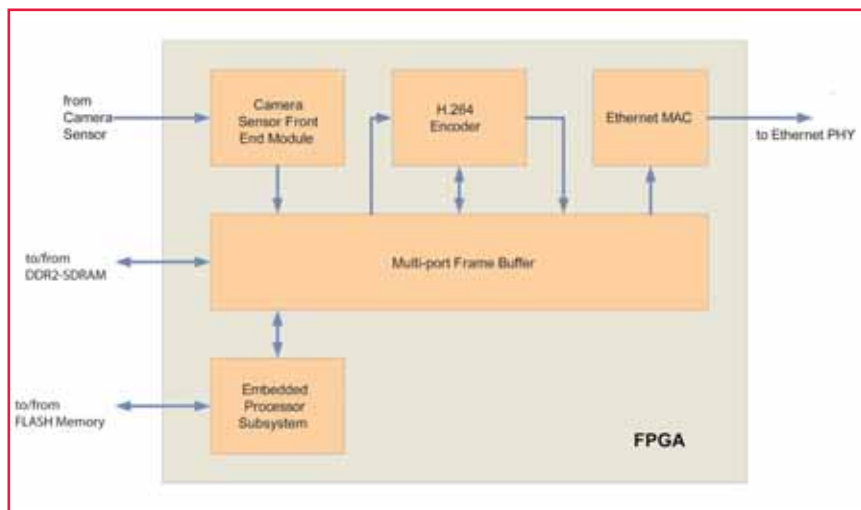


Fig. 1 – Architettura ad alto livello di un sistema di sorveglianza ad alta definizione

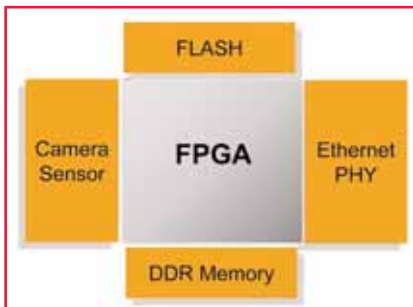


Fig. 2 – I vari componenti che formano la telecamera di sorveglianza HD

re integrate in un singolo dispositivo al fine di soddisfare i severi vincoli di costo e di dissipazione che contraddistinguono questo tipo di sistemi.

Nella figura 1 viene riportata la struttura ad alto livello di questa telecamera. Si può notare che oltre all'engine di codifica, un progetto che tenga conto del contenimento dei costi deve prevedere l'inclusione dei circuiti per la pre-elaborazione delle informazioni provenienti dal sensore della telecamera, il controllore di memoria per il frame buffer, un processore embedded per il controllo del sistema e un controllore MAC Ethernet. Obiettivo del progetto è ottenere il livello più spinto di integrazione possibile e un rapporto ottimizzato in termini del rapporto tra costo e consumi per l'intero sistema.

Come visibile nella figura 1 il progetto contempla un modulo di front end per il sensore della telecamera, un modulo per la compressione video, un modulo MAC Ethernet, un processore integrato e un frame buffer a più porte che mette a disposizione le risorse di memorizzazione necessarie per gli altri moduli.

Il frame buffer multi porta si comporta come un hub. Tutti gli altri moduli inviano e ricevono i dati verso/dal frame buffer il quale comunica con gli altri moduli. Le immagini video ricevute dal senso-

re della telecamera vengono inviate al modulo di front end del sensore.

Quest'ultimo elabora i dati video e memorizza il video nel frame buffer. Successivamente l'encoder H.264 legge questi dati video dal frame buffer ed effettua il processo di codifica. A questo punto l'encoder memorizza il bit stream compresso nel frame buffer. Il modulo Ethernet MAC, infine, legge il bit stream compresso dal frame buffer e lo invia sfruttando la rete Ethernet.

Grazie un livello di integrazione così elevato, i soli componenti presenti sulla scheda sono il sensore della telecamera, le memorie SDRAM-DDR2 e la memoria flash e il chip per il livello fisico (PHY) Ethernet. Nella figura 2 sono riportate i differenti componenti di questo sistema.

sinistra dello schema di figura 3 legge le immagini prelevate dal frame buffer seguendo un determinato percorso di scansione (raster) e le riorganizza in un formato a macroblocco. Il macroblocco viene quindi inviato all'engine per la stima del moto (Motion Estimate Engine) e all'engine per la stima spaziale (Spatial Estimation Engine). Il primo engine legge un'immagine di riferimento dal frame buffer e trova il vettore di moto del macroblocco attuale mediante una ricerca nell'immagine di riferimento. L'engine per la stima del moto determina inoltre la partizione ottimale per ogni macroblocco.

Le specifiche H.264 prevedono quattro differenti partizioni del macroblocco per le predizioni inter-frame e altrettante per i sotto-macroblocchi. Il vettore di

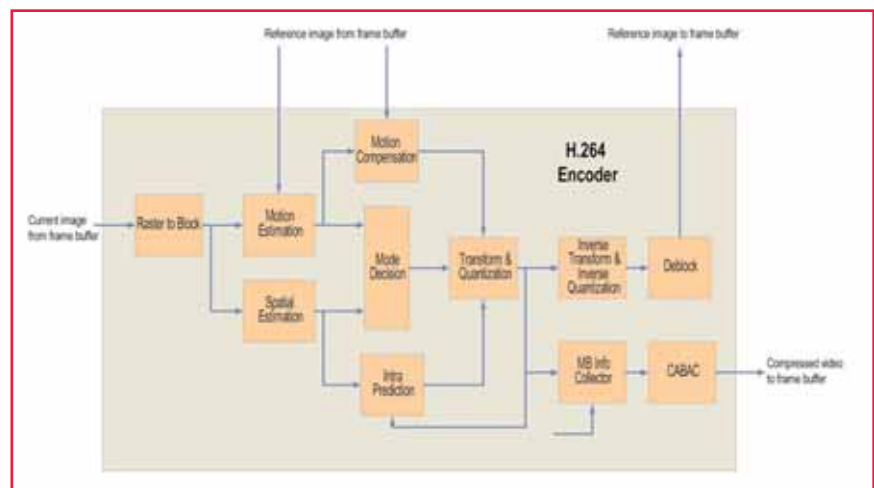


Fig. 3 – Architettura del codificatore H.264

Il modulo encoder H.264

L'encoder utilizzato in questo progetto è un core IP sviluppato da EyeLytics e ottimizzato da quest'ultima per applicazioni di sorveglianza. Il core (la cui architettura è riportata in Fig. 3) integra parecchie risorse tipiche delle applicazioni di sorveglianza - supporto di più canali, controllo della velocità in modo da garantire una qualità ostante, modalità inter/intra frame, QPEL, CABAC - e un numero limitato di gate.

Il modulo "Raster to Block" visibile a

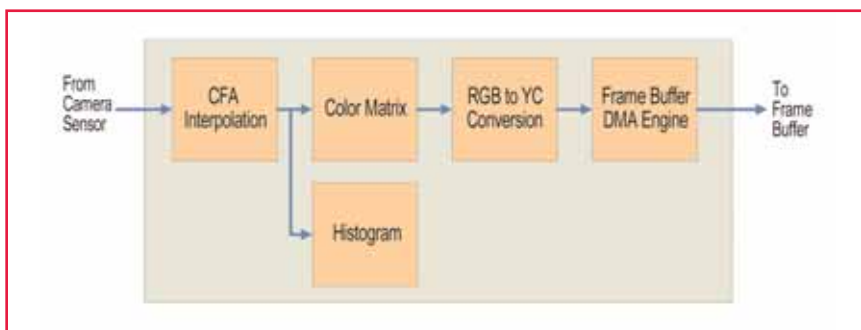


Fig. 4 – Schema a blocchi del front end del sensore della telecamera

movimento e la partizione ottimali insieme al costo di codifica del macroblocco stimato corrispondente vengono inviati al modulo per la decisione della modalità (Mode Decision Module). Le informazioni relative al vettore di movimento e alla partizione migliori vengono anche trasferite all'engine per la compensazione del movimento. Quest'ultimo preleva le regioni di riferimento corrispondenti, esegue il filtraggio su $\frac{1}{2}$ pixel (hal pel) e $\frac{1}{4}$ di pixel (quarter pel) e genera la predizione inter frame per l'attuale macroblocco.

L'engine di stima spaziale individua la stima migliore dell'attuale macroblocco utilizzando i valori dei pixel adiacenti all'interno della stessa immagine. Per quanto riguarda le predizioni intra, le specifiche H.264 definiscono due modalità per la predizione della componente di luminanza (luma): Intra 4x4 (sono possibili nove diversi modi di predizione) e Intra 16 x 16 (4 possibili modi di predizioni). La predizione intra dei segnali di crominanza (chroma) è sempre fatta su blocchi 8 x 8 (in questo caso sono possibili quattro predizioni).

L'engine di stima spaziale determina le migliori modalità di luminanza e crominanza da utilizzare per l'attuale macroblocco. Le migliori modalità insieme al

costo di codifica del macroblocco stimato corrispondente vengono inviati al Mode Decision Module. Quest'ultimo confronta i costi di codifica del macroblocco e determina la predizione ottimale da utilizzare. Questa predizione può essere di tipo sia inter sia intra.

La migliore modalità intra viene inviata all'Intra Prediction Engine che utilizza i valori dei pixel adiacenti per generare la predizione intra per l'attuale macroblocco.

Sulla base dei risultati del Mode



Fig. 5 - Il sistema di sorveglianza completo basato sulla piattaforma di sviluppo Cyclone III di Altera

Decision Module, l'engine di trasformazione e quantizzazione (Transform & Quantization) sottrae i valori di predizione dall'attuale macroblocco. A questo punto esso effettua le operazioni di trasformazione e quantizzazione per generare i coefficienti quantizzati. Questi coefficienti insieme ad altre informazioni relative al macroblocco – come ad esempio vettore di movimento, modalità di previsione inter e intra – vengono raccolte dal corrispondente macroblocco (Macroblock Info Collector).

Tutte le informazioni acquisite sono trasferite al modulo CABAC (Context Adaptive Binary Arithmetic Coding – codifica aritmetica binaria sensibile al contesto) per generare il bit stream finale che viene quindi inviato al frame buffer.

Le immagini di riferimento utilizzate dagli engine per la stima del movimento e la compensazione del movimento sono generate in tre fasi: quantizzazione inversa, trasformazione inversa e deblocking. Il Deblock Module esegue una particolare operazione di filtraggio in modo da ridurre gli artefatti (in pratica una distorsione prodotta dall'uso di tecniche di compressione molto spinte) dell'immagine. Le immagini elaborate vengono a questo punto inviate al frame buffer e impiegate come immagini di riferimento per il periodo di frame successivo.

Modulo di front end del sensore della telecamera

Il sensore della telecamera utilizzato in questo progetto è MT9P031, un sensore da 5 Megapixel sviluppato da Aptina. Il sensore è programmato in modo da inviare immagini video con risoluzione pari a 1920 x 1080 all'FPGA a una velocità di 30 fps (frame per second). Il formato dei dati video utilizzato è il pattern di Bayer RGB a 12 bit. Il clock per il trasferimento dei dati è impostato a una frequenza di 100 MHz.

Nella figura 4 viene riportata l'interpolazione

zione CFA utilizzata per convertire il pattern di Bayer RGB in un'immagine RGB regolare. La matrice di colore viene utilizzata per regolare il colore RGB per adattarlo a quello del mondo reale.

Successivamente viene effettuata la conversione da RGB a YC che genera i dati video in formato YC 4:2:2: questi ultimi vengono poi salvati nel frame buffer mediante l'engine DMA.

Altri componenti del progetto

Un processore embedded viene utilizzato per la programmazione dei vari registri all'interno dei differenti moduli come pure per far girare lo stack TCP/IP per lo streaming del video compresso. Operando con il modulo MAC Ethernet il processore fa girare IwIP (un'implementazione "leggera" dello stack TCP/IP), l'applicazione per lo streaming e un'applicazione per il web server.

Il frame buffer multi porta si collega a due chip SDRAM-DDR2 con un bus dati a 32 bit e opera a 150 MHz. L'ampiezza di banda di memoria massima è dunque uguale a 1,2 Gbps.

Il sistema completo

Il sistema completo basato su FPGA è riportato in figura 5. Come si può vedere, il sistema risulta formato da tre schede: una scheda di sviluppo che contiene un componente della serie Cyclone III di Altera – precisamente il dispositivo 3C120, un modulo sensore da 5 Mpixel Aptina e un adattatore EyeLytics che collega il sensore Aptina con la piattaforma di sviluppo basata sull'FPGA.

È altresì disponibile un progetto dimostrativo che mostra la codifica delle immagini del sensore della telecamera e la trasmissione in modalità streaming attraverso la rete a un PC.

L'intero progetto richiede circa 50.000 elementi logici (LE) e il modulo encoder utilizza circa 30.000 LE quando si vuole implementare la funzione di codifica H.264 Main Profile per un flusso video 720p/30.

I vantaggi degli FPGA

I sistemi di sorveglianza sono molto diversi tra loro, specialmente quelli ad alta definizione. I sistemi progettati per la sicurezza delle frontiere sono molto diversi da quelli impiegati nei casinò, anche se entrambi i tipi sono equipaggiati con sen-

sori ad alta definizione. Un'architettura basata su FPGA, se da un lato risulta estremamente flessibile e personalizzabile, dall'altra può essere implementata su una piattaforma di silicio standard e ampiamente collaudata. L'architettura del sistema descritto in questo articolo può essere migliorata e modificata per soddisfare esigenze differenti tra loro. Operazioni quali l'aumento della risoluzione della telecamera, l'aggiunta di funzioni di elaborazione video personalizzate e di un engine per l'analisi video risultano estremamente semplici in quanto il progetto è realizzato sfruttando un linguaggio HDL standard. Il modulo encoder H.264 supporta la codifica su più canali e flussi con risoluzioni differenti, in modo da consentire la codifica contemporanea di video con diverse frame rate. Poiché il progetto può essere implementato su qualsiasi FPGA, l'adozione di un approccio di progettazione aperto consente di utilizzare gli FPGA più recenti, in modo da ottenere prestazioni sempre migliori e ottimizzare ancora di più il rapporto tra costi e consumi.

readerservice@fieramilanoeditore.it

Altera
EyeLytics

n.6
www.eyelytics.com