



# Sistemi di visione ad alte prestazioni supportati da FPGA

**Basler Italy S.r.l.**  
Matteo Seregni

Promosso da



Organizzato da



## Sistemi di visione ad alte prestazioni

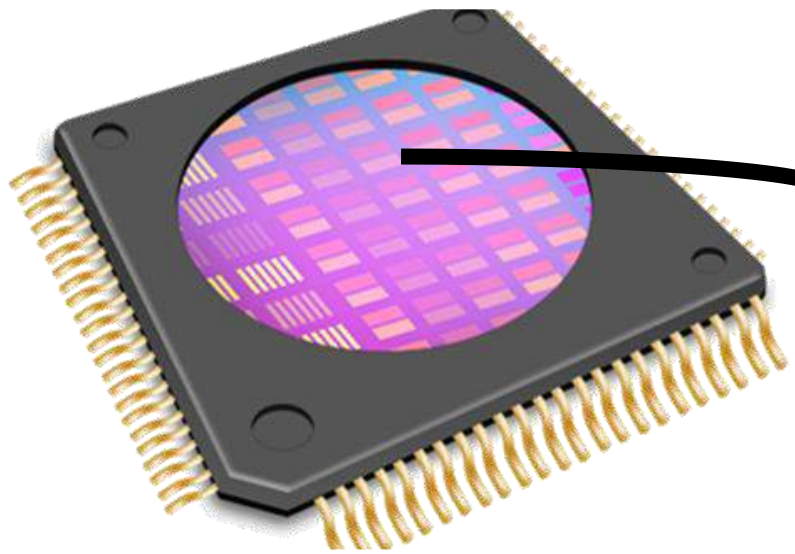


Real time  
Bassa latenza (ms ~  $\mu$ s)



Alti FPS  
Alta risoluzione  
Banda elevata (~ GB/s)

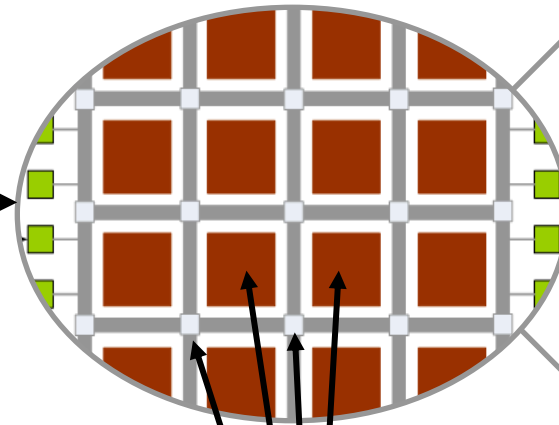
## FPGA (*Field Programmable Gate Array*)



**FPGA**

**Circuito digitale programmabile**

*(senza interventi fisici)*



**Matrice di blocchi  
logici interconnessi**

**Programmazione**

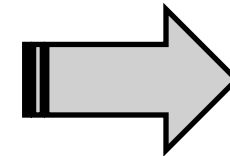
## CPU/GPU vs FPGA

### CPU / GPU

- Il task da eseguire è definito da un software, che viene scomposto in una serie di istruzioni di basso livello.
- Ogni *core* esegue le **istruzioni in sequenza** (una per ogni colpo di clock)

### FPGA

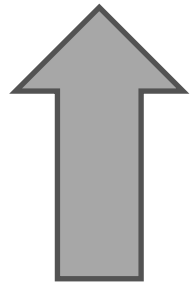
- Il task da eseguire viene tradotto in una configurazione hardware
- Un FPGA programmato è un circuito digitale dedicato al task specifico.
- Ad ogni colpo di clock (tick), il circuito esegue tutte le **operazioni in parallelo**



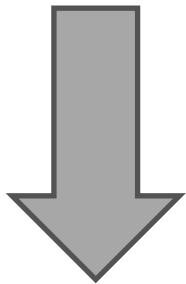
- **Banda elevata**
- **Bassa latenza**
- **Tempo di risposta deterministico**

# Programmazione FPGA 'tradizionale'

**Linguaggi HDL**  
(*Hardware description language*)



Massima flessibilità



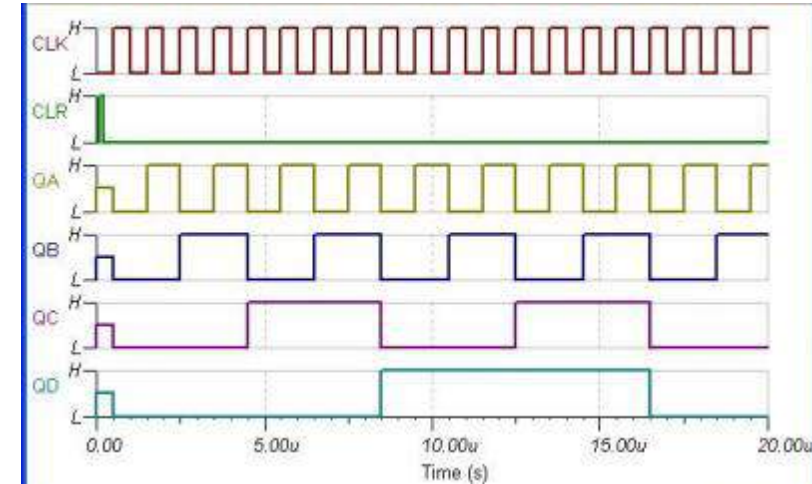
Complessità  
(sviluppo e testing)

Per esperti

```

1 library ieee;
2 use ieee.std_logic_1164.all;
3 use ieee.numeric_std.all;
4
5 entity signed_adder is
6 port
7 (
8   aclr : in  std_logic;
9   clk  : in  std_logic;
10  a    : in  std_logic_vector;
11  b    : in  std_logic_vector;
12  q    : out std_logic_vector
13 );
14 end signed_adder;
15
16 architecture signed_adder_arch of signed_adder is
17   signal q_s : signed(a'high+1 downto 0); -- extra bit wide
18
19 begin -- architecture
20   assert(a'length >= b'length)
21     report "Port A must be the longer vector if different sizes!"
22     severity FAILURE;
23   q <= std_logic_vector(q_s);
24
25   adding_proc:
26   process (aclr, clk)
27     begin
28       if (aclr = '1') then
29         q_s <= (others => '0');
30       elsif rising_edge(clk) then
31         q_s <= ('0'&signed(a)) + ('0'&signed(b));
32       end if; -- clk'd
33     end process;
34
35 end signed_adder_arch;

```



## Programmazione FPGA per la visione

Programmazione ad  
alto livello

- ✓ HDL «nascosto» da  
interfacce grafiche (GUI) o  
testuali (SDK)

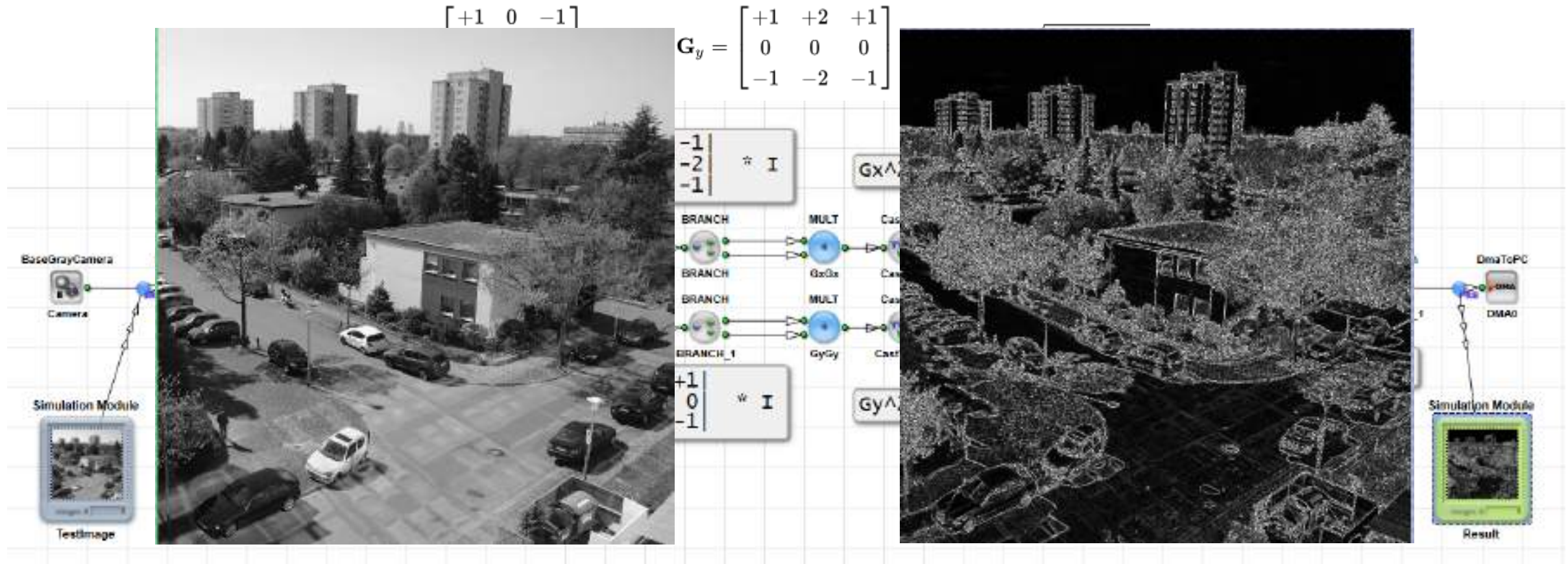
Strumenti dedicati  
alla visione

- ✓ Librerie di operatori per  
image processing
- ✓ Simulazione basata su  
immagini

**Accessibilità**

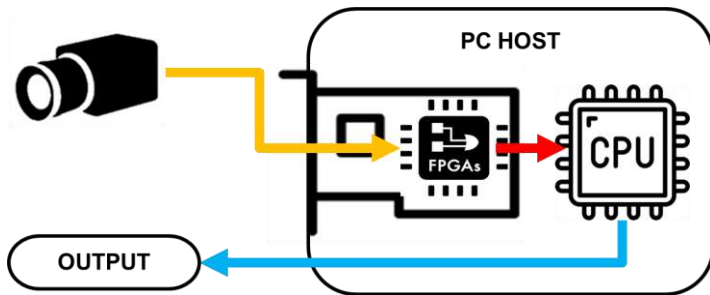
- ✓ Competenze specifiche su  
FPGA non necessarie
- ✓ Minori tempi/costi di sviluppo

## Programmazione FPGA per la visione: esempio GUI



## Utilizzo FPGA nei sistemi di visione

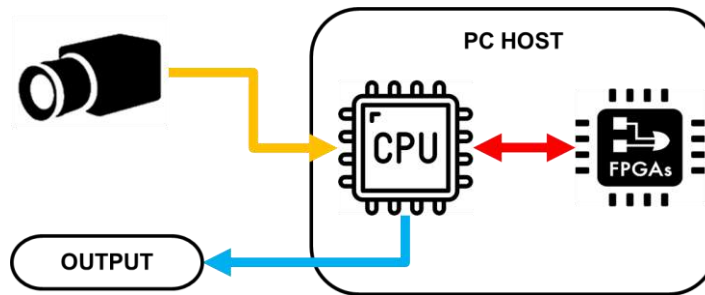
### PreProcessing



- Preelaborazione su FPGA
- Elaborazione su CPU

✓ Sinergia FPGA - CPU

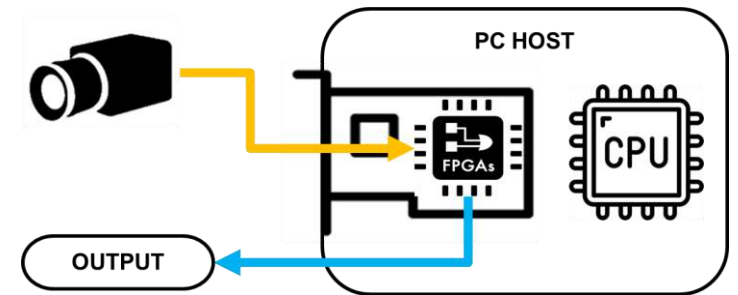
### CoProcessing



- CPU delega parte del carico all'FPGA
- FPGA = acceleratore HW

✓ Sinergia FPGA - CPU

### Processing Esclusivo



- Elaborazione solo su FPGA
- CPU scarica

✓ Latenza minima possibile



## Integrazione FPGA nei sistemi di visione



**Modalità di  
utilizzo**

PreProcessing  
CoProcessing  
Processing esclusivo

CoProcessing

PreProcessing  
Processing esclusivo

**Interfaccia di  
acquisizione**

CL, CoaXPress,  
(10)GigE

Qualsiasi

Qualsiasi  
(GigE e USB3)

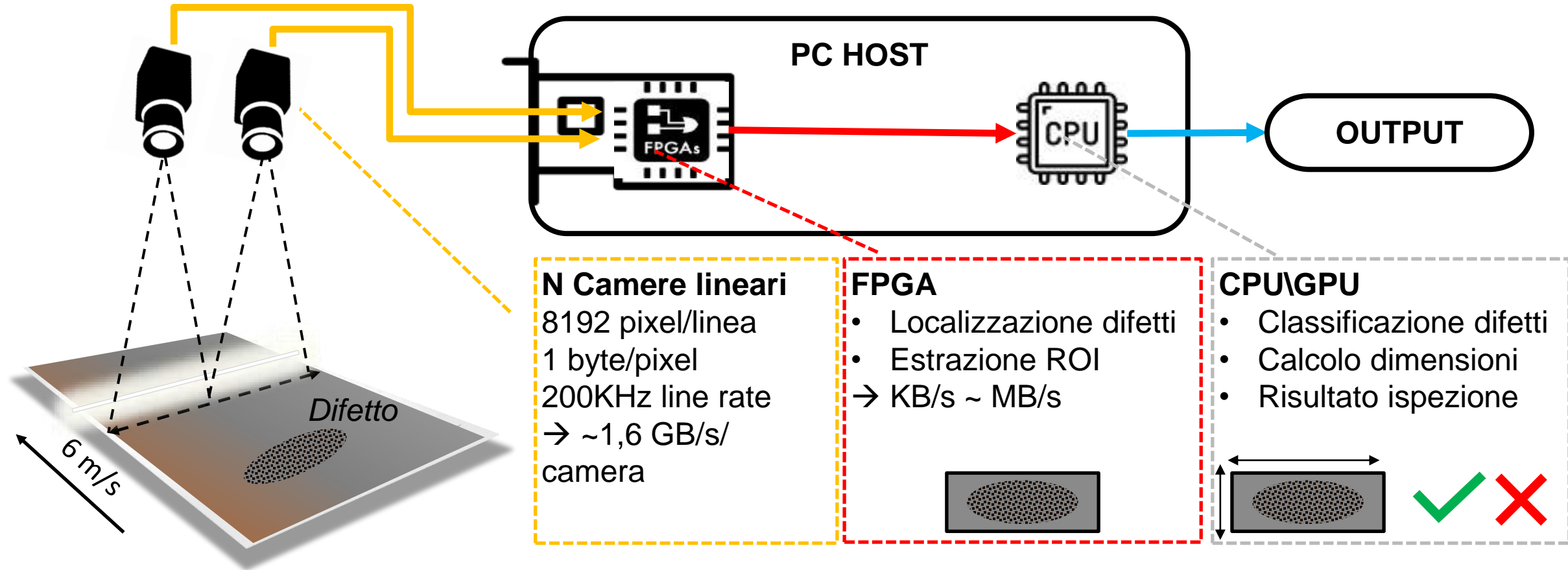
**Componenti  
aggiuntivi?**

NO (CL o CoaX)  
SI (GigE)

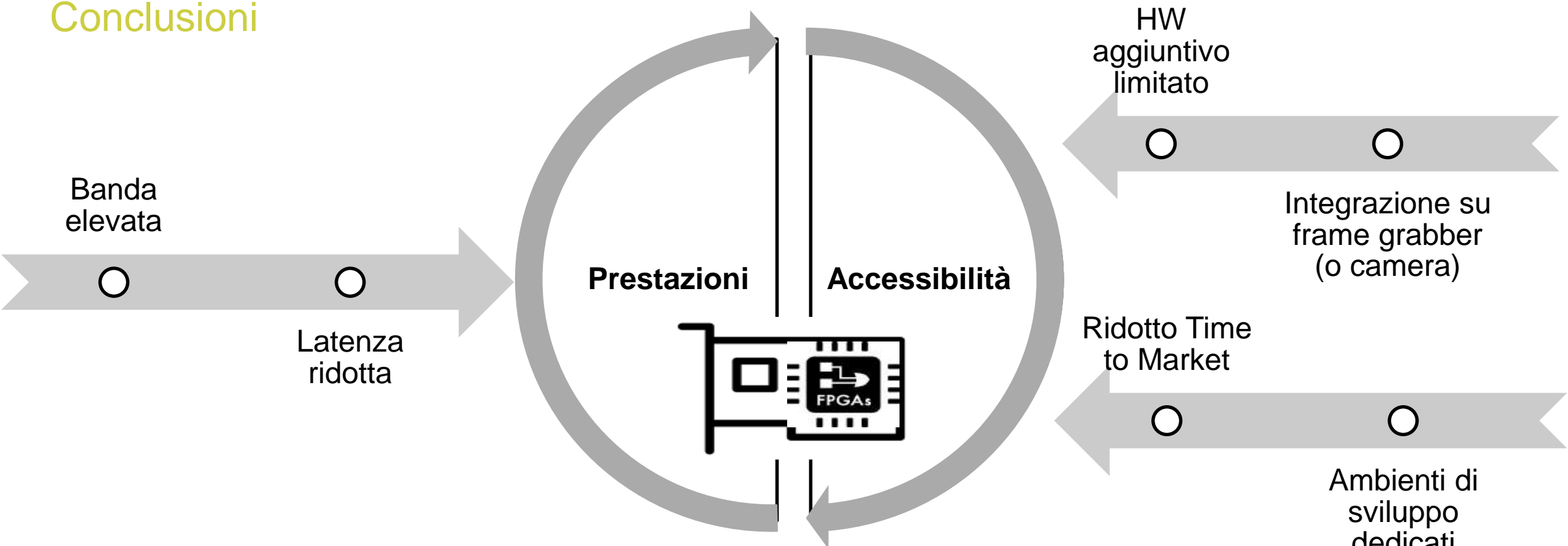
SI

NO

## Esempio: ispezione rivestimento elettrodi nella produzione di batterie



## Conclusioni





ITALIAN MACHINE  
VISION FORUM

**GRAZIE PER L'ATTENZIONE**

Promosso da



Organizzato da



## Contatti



### Matteo Seregni

*Senior Computer Vision Software Engineer*  
Basler Italy

[matteo.seregni@baslerweb.com](mailto:matteo.seregni@baslerweb.com)



[marketing.italy@baslerweb.com](mailto:marketing.italy@baslerweb.com)



[basler-italy](https://www.linkedin.com/company/basler-italy)



[baslerweb.it](https://www.baslerweb.it)

**BASLER**