Høgskulen
på Vestlandet

ELE111 Digitale design

F11_000 Ekstern S-ram

Eivind Skjæveland
esk@hvl.no

SRAM på DE2-115-kortet

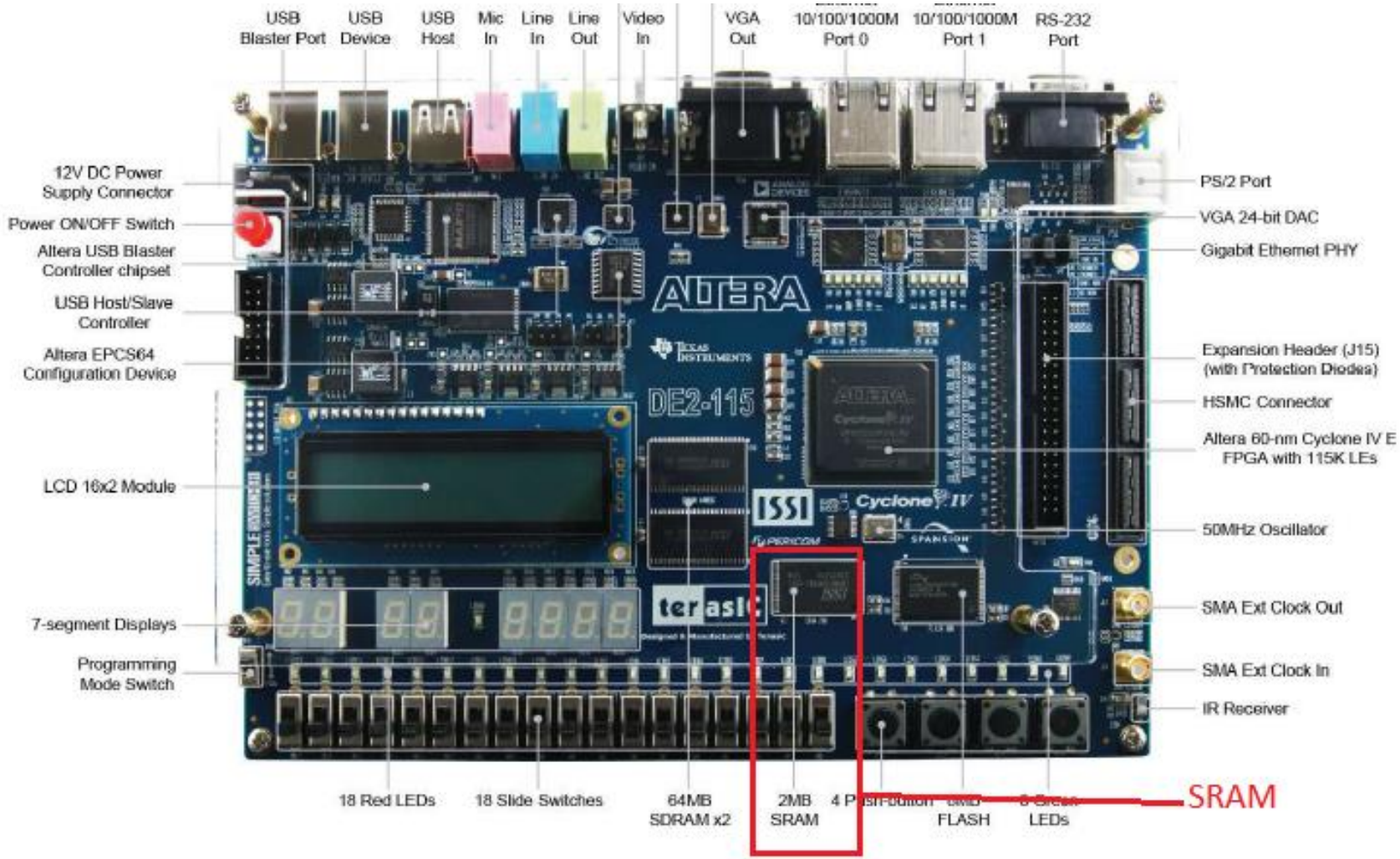


Figure 2-1 The DE2-115 board (top view)

SRAM på DE2-115-kortet

- SRAM-krets på kortet
  - IS61WV102416ALL
  - Statisk RAM
  - 1024 16-bits ord
  - 20-bits adresseport
  - 16bits toveis dataport
  - 5 bit kontrollerbuss

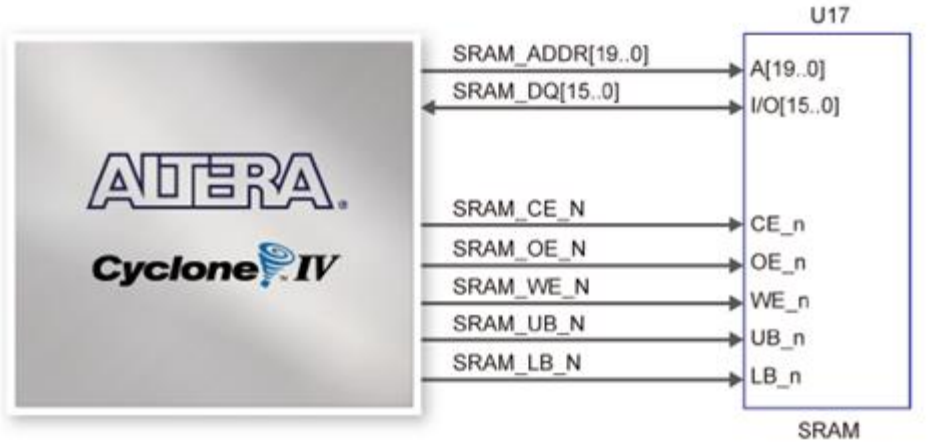


Figure 4-33 Connections between FPGA and SRAM

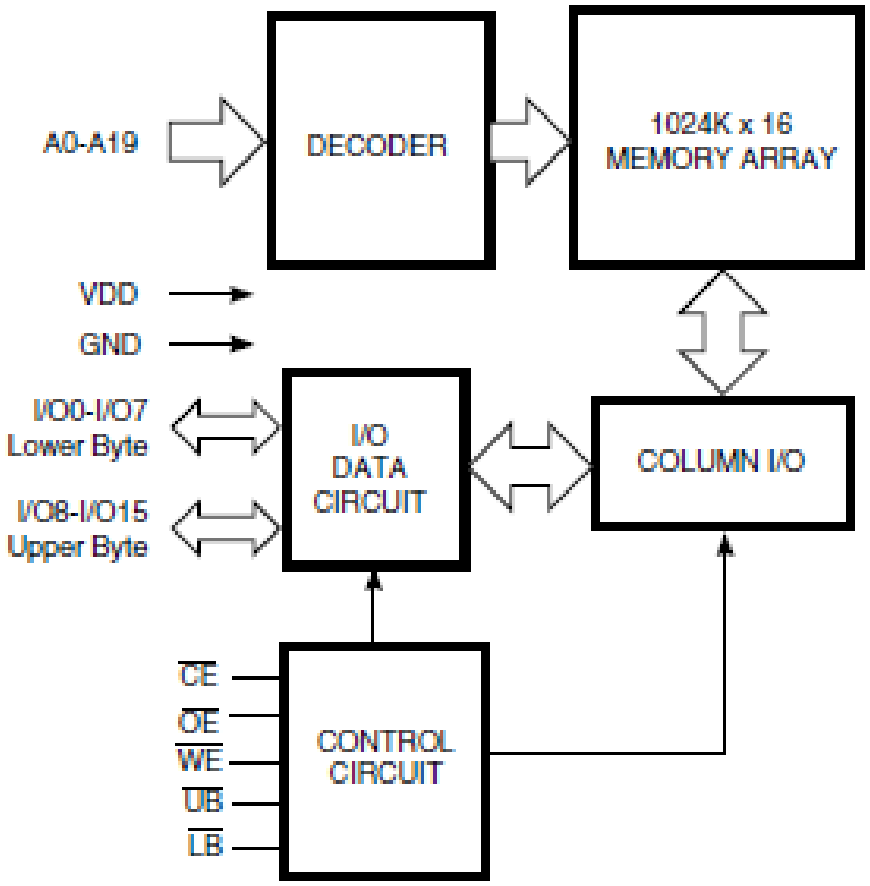
SRAM IS61WV102416ALL

- Pinner på kretsen

PIN DESCRIPTIONS

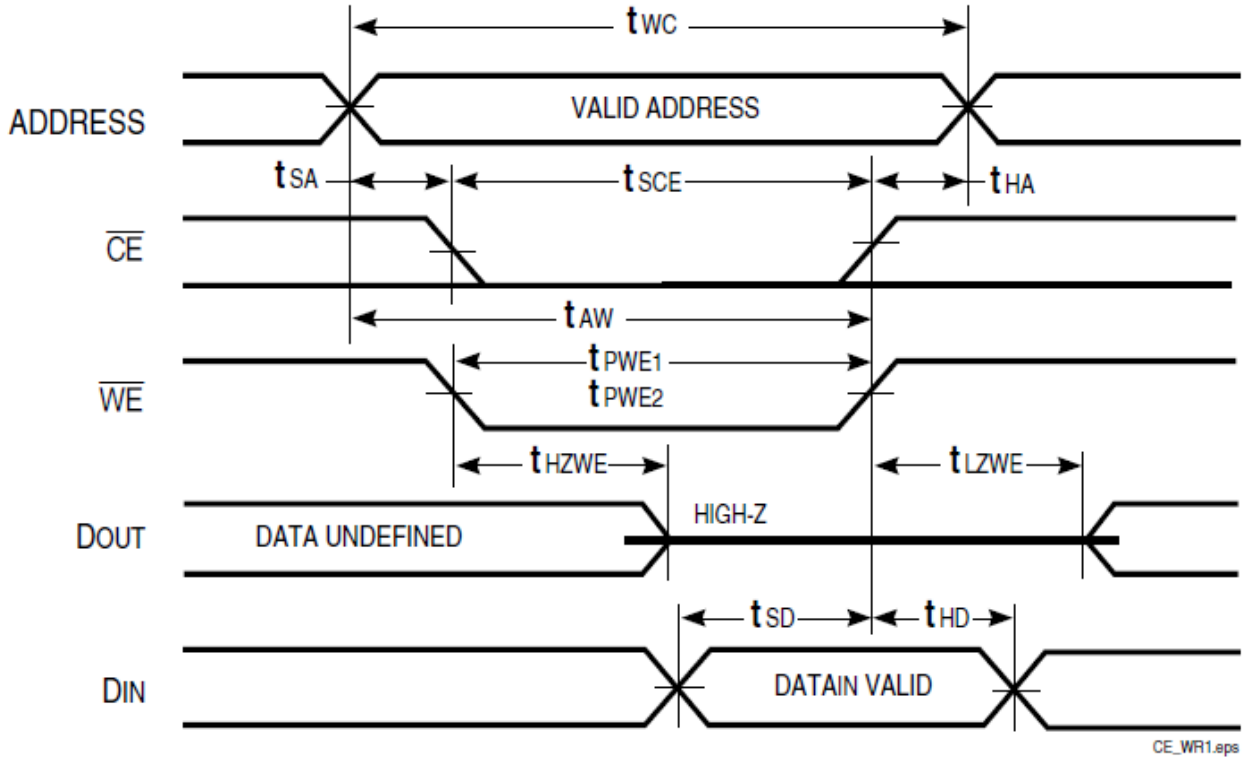
A0-A19	Address Inputs
I/O0-I/O15	Data Inputs/Outputs
$\overline{CE}$	Chip Enable Input
$\overline{OE}$	Output Enable Input
$\overline{WE}$	Write Enable Input
$\overline{LB}$	Lower-byte Control (I/O0-I/O7)
$\overline{UB}$	Upper-byte Control (I/O8-I/O15)
NC	No Connection
V_{DD}	Power
GND	Ground

- Blokkindeling



Skrive-syklus

1. Velg adresse
2. CE = 0, chip select
3. WE = 0, Write enable
4. Setup-tid
   - Data må vera i ro i 5 ns, t_{SD}, før WE går høg
   - Adresse må vera i ro i 6,5 ns, t_{AW} før WE går høg
5. Hold –tid
   - Tida data og adresser må vera i ro etter at skrivesyklus er ferdig
   - 0 ns for denne kretsen



Timing-parameter for skriving

WRITE CYCLE SWITCHING CHARACTERISTICS[1,3] (Over Operating Range)

Symbol	Parameter	-8 Min.	-8 Max.	-10 Min.	-10 Max.	Unit
t_{WC}	Write Cycle Time	8	—	10	—	ns
t_{SCE}	$\overline{CE}$ to Write End	6.5	—	8	—	ns
t_{AW}	Address Setup Time to Write End	6.5	—	8	—	ns
t_{HA}	Address Hold from Write End	0	—	0	—	ns
t_{SA}	Address Setup Time	0	—	0	—	ns
t_{PWB}	$\overline{LB}$, $\overline{UB}$ Valid to End of Write	6.5	—	8	—	ns
t_{PWE1}	$\overline{WE}$ Pulse Width	6.5	—	8	—	ns
t_{PWE2}	$\overline{WE}$ Pulse Width ($\overline{OE}$ = LOW)	8.0	—	10	—	ns
t_{SD}	Data Setup to Write End	5	—	6	—	ns
t_{HD}	Data Hold from Write End	0	—	0	—	ns
t_{HZWE}[2]	$\overline{WE}$ LOW to High-Z Output	—	3.5	—	5	ns
t_{LZWE}[2]	$\overline{WE}$ HIGH to Low-Z Output	2	—	2	—	ns

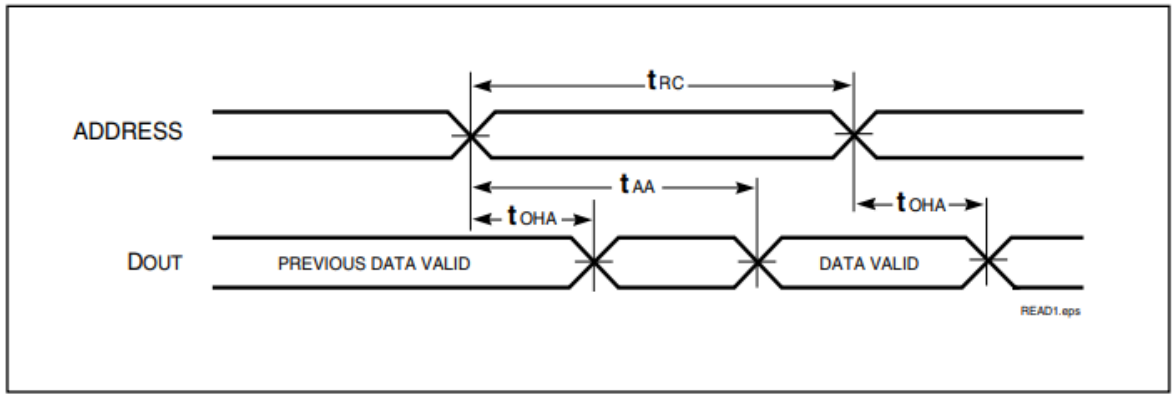
Notes:

1. Test conditions assume signal transition times of 3 ns or less, timing reference levels of 1.5V, input pulse levels of 0V to 3.0V and output loading specified in Figure 1.
2. Tested with the load in Figure 2. Transition is measured ±500 mV from steady-state voltage. Not 100% tested.
3. The internal write time is defined by the overlap of CE LOW and UB or LB, and WE LOW. All signals must be in valid states to initiate a Write, but any one can go inactive to terminate the Write. The Data Input Setup and Hold timing are referenced to the rising or falling edge of the signal that terminates the write. Shaded area product in development

Lese-syklus

- Kan lesa når WE = 1
  - OE =0, CE = 0
- Når adresse blir endra vil data vera klare på Dout etter 20 ns, t_{AA}

AC WAVEFORMS
READ CYCLE NO. 1[1,2] (Address Controlled) ($\overline{CE}$ = $\overline{OE}$ = V_{IL})



timing-parameter for lesing

READ CYCLE SWITCHING CHARACTERISTICS[1] (Over Operating Range)

Symbol	Parameter	-20 ns Min.	-20 ns Max.	Unit
t_{RC}	Read Cycle Time	20	—	ns
t_{AA}	Address Access Time	—	20	ns
t_{OHA}	Output Hold Time	2.5	—	ns
t_{ACE}	$\overline{CE}$ Access Time	—	20	ns
t_{DOE}	$\overline{OE}$ Access Time	—	8	ns
t_{HZOE}[2]	$\overline{OE}$ to High-Z Output	0	8	ns
t_{LZOE}[2]	$\overline{OE}$ to Low-Z Output	0	—	ns
t_{HZCE}[2	$\overline{CE}$ to High-Z Output	0	8	ns
t_{LZCE}[2]	$\overline{CE}$ to Low-Z Output	3	—	ns
t_{BA}	$\overline{LB}$, $\overline{UB}$ Access Time	—	8	ns
t_{HZB}	$\overline{LB}$, $\overline{UB}$ to High-Z Output	0	8	ns
t_{LZB}	$\overline{LB}$, $\overline{UB}$ to Low-Z Output	0	—	ns

Val av kontrollsignal

- Kan konfigurera RAM-en på mange ulike måtar ved hjelp av kontrollsignala.
- I LAB 3 gjer vi det enkelt
  - CE, OE, UB ,LB = , kobla til jord
  - Aktiv låg signal
    - Ramen alltid tilgjengeleg, kan lesast så lenge vi ikkje er i skrivesyklus
      - WE = 0, Write enable

TRUTH TABLE

Mode	$\overline{WE}$	$\overline{CE}$	$\overline{OE}$	$\overline{LB}$	$\overline{UB}$	I/O PIN I/O0-I/O7	I/O PIN I/O8-I/O15	Vcc Current
Not Selected	X	H	X	X	X	High-Z	High-Z	I_{SB1}, I_{SB2}
Output Disabled	H	L	H	X	X	High-Z	High-Z	I_{CC}
	X	L	X	H	H	High-Z	High-Z	
Read	H	L	L	L	H	D_{OUT}	High-Z	I_{CC}
	H	L	L	H	L	High-Z	D_{OUT}	
	H	L	L	L	L	D_{OUT}	D_{OUT}	
Write	L	L	X	L	H	D_{IN}	High-Z	I_{CC}
	L	L	X	H	L	High-Z	D_{IN}	
	L	L	X	L	L	D_{IN}	D_{IN}	

Entity for tilkoling til ekstern RAM

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity L3_SRAM is
    port(
    SW          : in std_logic_vector(17 downto 0);
    KEY         : in std_logic_vector(3 downto 0);
    LEDR        : out std_logic_vector(17 downto 0);
    SRAM_ADDR   : out std_logic_vector(19 downto 0);
    SRAM_DQ     : inout std_logic_vector(15 downto 0);
    SRAM_WE_N   : buffer std_logic;
    SRAM_CE_N, SRAM_OE_N, SRAM_UB_N, SRAM_LB_N  : out
std_logic;
    HEX7,HEX6,HEX5,HEX4, HEX3,
    HEX2,HEX1,HEX0  : out std_logic_vector(6 downto 0)
    );
end entity L3;
```

kontrollsignal

```
SRAM_CE_N <='0';
SRAM_OE_N <='0';
SRAM_UB_N <='0';
SRAM_LB_N <='0';
SRAM_WE_N <= SW(17);
```

Data og adresse

```
    adr <= SW (15 downto 11);
    DataINN <= SW(7 downto 0);

--Databussen
   -- bruker berre 8 bit data, legg 8 MSB i databussen til '0'
   -- under lesing (WE_N = '0') skal datautgangen vera tristate.
   SRAM_DQ <= "000000000" & DataInn when SRAM_WE_N = '0' -- data in til SRAM
              else (others => 'Z');
   DataUT <= SRAM_DQ(7 downto 0);  --data ut frå SRAM

   -- legg 0 til dei mest signifikante bita i addressa
   SRAM_ADDR(19 downto 5) <= (others => '0');
   SRAM_ADDR(4 downto 0) <= adr;
```

Data til og frå SRAM

- Vi brukar same pinnane til å senda data inn og ut frå S-RAM
  - SRAM_DQ er inout-port
  - WE styrer om vi skal lesa eller skriva til SRAM:
  - Sett utgangen til tri-state ('Z') når vi ikkje skal skriva til SRAM.

```
--Databussen
    -- bruker berre 8 bit data, legg 8 MSB i databussen til '0'
    -- under lesing (WE_N = '0') skal datautgangen vera tristate.
    SRAM_DQ <= "000000000" & DataInn when SRAM_WE_N = '0' -- data in til SRAM
                else (others => 'Z');
    DataUT <= SRAM_DQ(7 downto 0);  --data ut frå SRAM
```

Kopling til 7-segment-display

```
tall0 : ROM_7_seg PORT MAP(adresse => DataUT(3 downto 0),HEX=> HEX0);
tall1 : ROM_7_seg PORT MAP(adresse => DataUT(7 downto 4),HEX=> HEX1);
HEX2 <= "1111111";-- av
HEX3 <= "1111111";-- av
tall4 : ROM_7_seg PORT MAP(adresse => DataINN(3 downto 0),HEX=> HEX4);
tall5 : ROM_7_seg PORT MAP(adresse => DataINN(7 downto 4),HEX=> HEX5);
tall6 : ROM_7_seg PORT MAP(adresse => adr(3 downto 0),HEX=> HEX6);
tall7 : ROM_7_seg PORT MAP(adresse => "000" & adr(4), HEX=> HEX7);
```

S-RAM