



K_BRD

DESCRIPTION – Keyboard controller

The K_BRD.VHD IP core, described here, is a matrix keyboard controller. It provides all the necessary logic to fully encode an array SPST switches. The keyboard scan is implemented with an internal 5KHz clock generated from the external clock input. No diodes in the switch array are needed to eliminate ghost swirches. The internal debounce logic, needs no external components and can be configured for any debounce interval. A Key Detect output goes low when a valid keyboard entry has been made. The Key Detect output returns to high level when the entered key is released unless another key is pressed before releasing the previous key. In other words, two key roll-over is not provided.. TRI-STATE outputs are provided for easy bus operation.

FEATURES

- **Double buffered interface logic** allows configuration to any bus width, from **serial** to **parallel** of any width.
- Interface on data bus or port, **dedicated or shared**.
- Instantiate the core as **stand-alone** or **cascaded**, in a daisy chain configuration.
- **Daisy chain** multiple **cores** on **one port address**.
- Configure for any number of **ROWS** and **COLUMNS**.
- Rising and falling edge **key debouncing**.
- Configurable **debounce interval**
- Configure for **registered or unregistered** column and scan code outputs to **minimize silicon usage**.
- Can be configured for **any clock frequency**

APPLICATION

The keyboard is the primary human interface tool in almost all industrial, commercial and consumer electronic products. The keyboard controller is used, not only to read devices which look like keyboards as in phones, microwave ovens, calculators, computers, oscilloscopes, wrist watches, TV and AC remotes, etc, but also operator consoles in large power plants and machines in factories, which do not look like keyboards. The switches in these consoles and keyboards are connected in a matrix and are read by a keyboard controller.



VHDL Component Declaration:

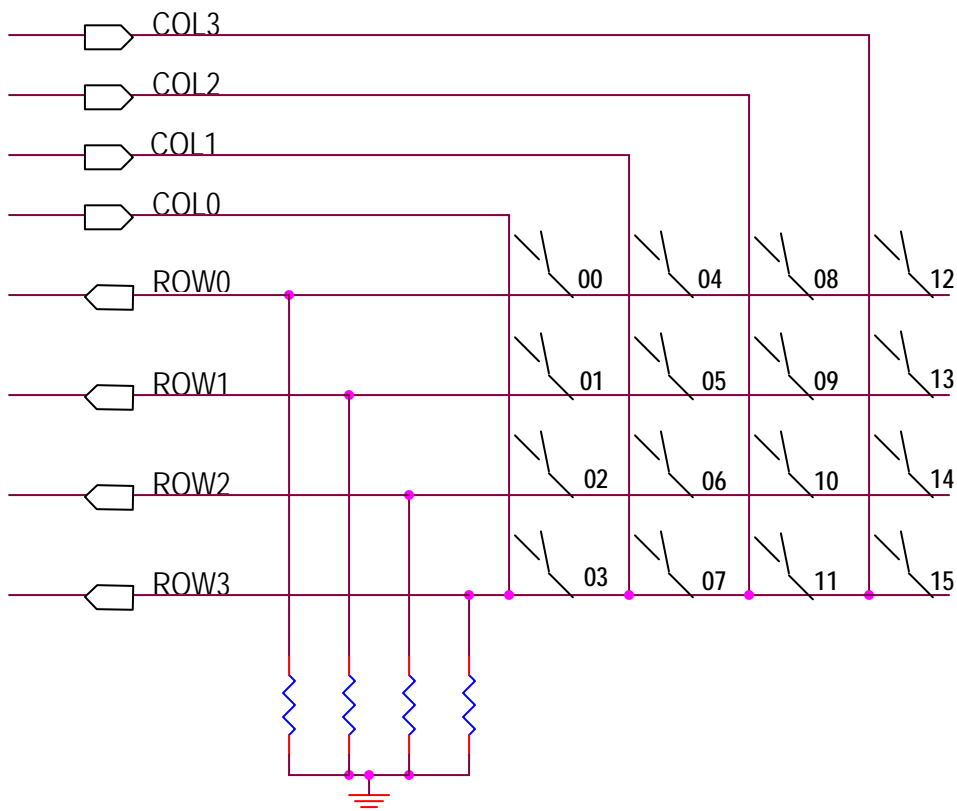
```
COMPONENT K_BRD
  GENERIC (
    IREG : INTEGER := 1;
    OREG : INTEGER := 1;
    KBD  : INTEGER := 1;
    NMR  : INTEGER := 4;
    NMC  : INTEGER := 4;
    DCN  : INTEGER := 0;
    DBW  : INTEGER := 16;
    CASC : INTEGER := 0;
    EBUS : INTEGER := 0;
    CEF  : INTEGER := 0;
    GCL  : INTEGER := 20000000
  );
  PORT(
    CLKI : IN  NODE;
    RST  : IN  NODE;
    ENB  : IN  NODE;
    ROW  : IN  BUS1D(NMR-1 DOWNT0 0);
    CS   : IN  NODE;
    MAST : IN  NODE;
    CASI : IN  NODE;
    D    : BUFFER BUS1D(DBW-1 DOWNT0 0);
    COD  : BUFFER BUS1D(NMC-1 DOWNT0 0);
    CASO : BUFFER NODE;
    CEN  : BUFFER NODE;
  );
END COMPONENT;
```

FILES YOU GET

i) FUNC.DOC	-	Documentation of functions & data types used in the core.
ii) README.DOC	-	Compile and licensing information.
iii) KBRD.DOC	-	This document
a) MYLIB.VHD	-	PACKAGE
b) K_BRD.VHD	-	TOP HIERARCHY DESIGN FILE
c) M_DFF.VHD	-	DESIGN FILE BELOW TOP HIERARCHY
d) S_DFF.VHD	-	-DO-
e) P_AD.VHD	-	-DO-
g) I_NCDEC.VHD	-	-DO-
h) U_DCNT.VHD	-	-DO-
i) A_DSB.VHD	-	-DO-
j) D_BIR.VHD	-	-DO-
k) S_TFF.VHD	-	-DO-
l) S_JKF.VHD	-	-DO-
m) F_DIV.VHD	-	-DO-
n) B_SHIFT.VHD	-	-DO-
o) D_ECOT.VHD	-	-DO-
p) P_LSE.VHD	-	-DO-
q) E_NCOT.VHD	-	-DO-
r) M_TRI.VHD	-	-DO-
s) S_TRI.VHD	-	-DO-
t) R_STK.VHD	-	-DO-



SAMPLE SCHEMATIC ARRANGEMENT



SCAN CODE		SWITCH LABEL IN SCHEMATIC	STICKER LABEL
MSB	LSB		
00	00	00	K1
00	01	01	K5
00	10	02	K9
00	11	03	K13
01	00	04	K2
01	01	05	K6
01	10	06	K10
01	11	07	K14
10	00	08	K3
10	01	09	K7
10	10	10	K11
10	11	11	K15
11	00	12	K4)
11	01	13	K8
11	10	14	K12
11	11	15	K16



INTERFACE INFORMATION

The interface schemes described here provide the user with information on how this core can be interfaced along with other peripherals on the common system resources, such as the data bus and input ports.

The two types of interface schemes are described here are:-

- 1) **Scan code unlatched - (stand-alone or cascaded)**
- 2) **Scan code latched - (stand-alone or cascaded)**

In the first interface scheme, the scan code is not latched by the core. While a key is pressed the 'key detect' bit (KD), of the scan code Goes Lo, indicating a valid scan code. The CPU program must latch the scan code when KD=0. When the key is released the KD bit will go Hi, indicating an invalid scan code.

In the second scheme, the scan code is latched internally by the core, when KD=0 and remains so, until it is read. If the key is released and another key is pressed, before the previous one is read, the previous scan code is overwritten. The CPU program, thus, needn't latch it.

The number of bits of scan code transmitted at a time depends on the number of data lines connected to the core. If the number of data lines connected is less than that required for a full parallel transmission, the scan code is sliced to the size of the data lines. The sliced scan code must be read contiguously until the whole data word is assembled by the CPU. Calculation of the number of slices of scan code (NSL) is shown below.

In all the interface schemes the data bus and port can be shared with any number of peripherals. However the Chip Select line (CS) must be unique for every core instantiated in the stand-alone scheme and common to all cascaded cores instantiated in the cascaded interface scheme. Chip-select signals can be generated either by an i/o address decoder, within the PLD(in the bus interface schemes) or from an output port on a micro-controller(in the port interface schemes).

The scan code data can be taken outside the PLD if the EBUS parameter is set to one, data is then routed through TRI-STATE buffers, which are enabled when the core is enabled with its CS input. The OEN (output enable) signal of these buffers is coincident with the CS input.

CASCADED

This option is enabled by setting the CASC parameter to one and connecting the MAST input of the first core in the chain to '1' and the CASO output of each core to the CASI input of the next core in the chain. The CASO output of the last core in the chain is connected to the CASI input of the master, or the first core in the chain.

In this scheme, although the chip select logic overhead is minimum, the cores cannot be accessed randomly and must be accessed sequentially one after the other and the entire sequence must be completed everytime, a process known as daisy chaining. The core, configured as the master will transmit first followed by the next in the chain. After a core transmits its data, it pulses the CASO output, which being connected to the next core in the chain, enables it for data transmission. After all the cores in the chain are read, the master is once again enabled.

STAND-ALONE

This option is enabled by setting the CASC parameter to zero, the MAST and CASI inputs are not used and may be connected to any constant.

In this scheme, the core is ready to transmit data soon after RST=1. The core with an active CS signal will send data and only one core at a time must be activated. Separate chip selects for each core bring with them the freedom of random access, however at the price of additional logic overhead and pinouts



CALCULATION OF NSL

$$A = \text{Log}_2(\text{NMR} - 1)$$

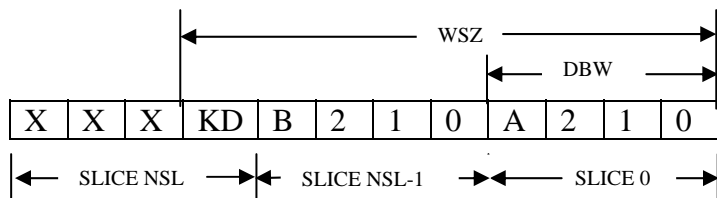
$$B = \text{Log}_2(\text{NMC} - 1)$$

$$\text{WSZ} = \text{SIZE OF DATA WORD TO BE READ} = A + B + 3$$

$$\text{NSL} = \lceil (\text{WSZ} + 1) / \text{DBW} \rceil - 1$$

If the result of division produces a remainder, add 1 to result

DATA WORD- IREG=1



EXAMPLE - For 16x16 keyboard, 4 bit data bus:-

$$\text{NMR} = 16$$

$$\text{NMC} = 16$$

$$\text{DBW} = 4$$

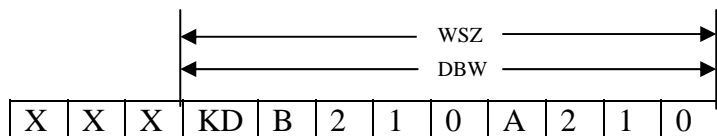
$$A = \text{Log}_2(16 - 1) = 3$$

$$B = \text{Log}_2(16 - 1) = 3$$

$$\text{WSZ} = 9$$

$$\text{NSL} = (8+1)/4 = (2 \text{ rem } 1) - 1 = 2$$

DATA WORD-IREG=0



EXAMPLE - For 16x16 keyboard, 4 bit data bus:-

$$\text{NMR} = 16$$

$$\text{NMC} = 16$$

$$A = \text{Log}_2(16 - 1) = 3$$

$$B = \text{Log}_2(16 - 1) = 3$$

$$\text{WSZ} = \text{DBW} = 9$$

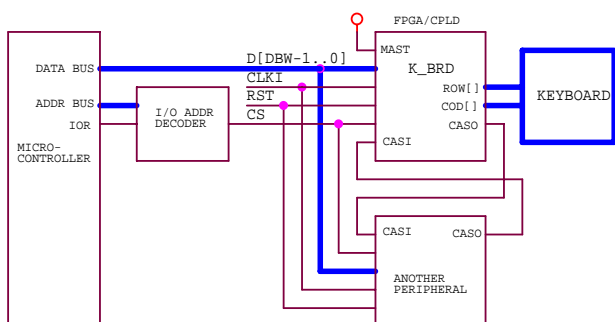
NSL = Not applicable



CIRCUIT DIAGRAMS

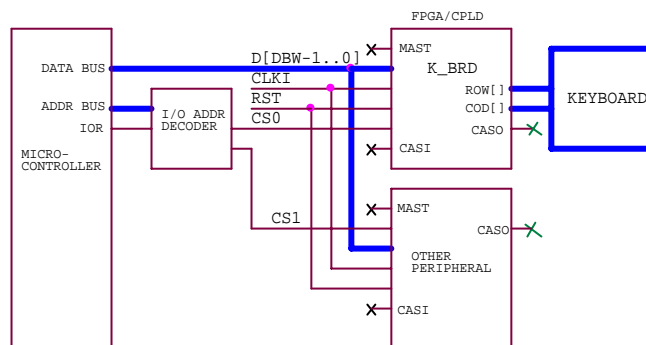
CASCADED EXTERNAL DATA BUS

Parameter CASC=1, EBUS=1



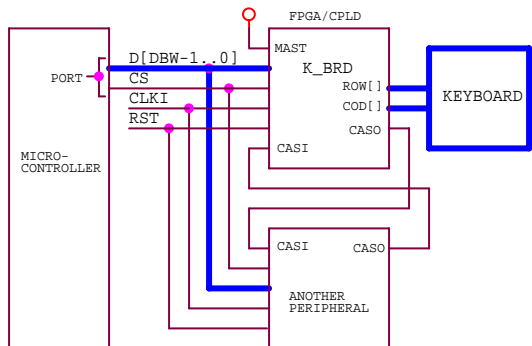
STAND-ALONE EXTERNAL DATA BUS

Parameter CASC=0, EBUS=1



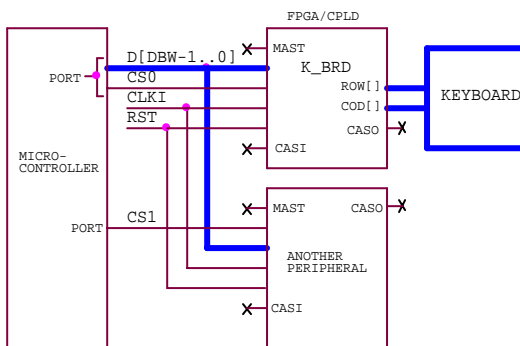
CASCADED EXTERNAL PORT

Parameter CASC=1, EBUS=1

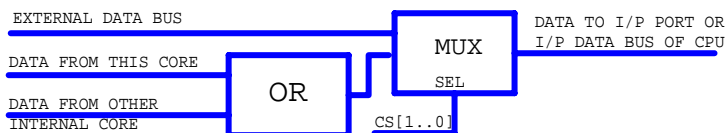


STAND-ALONE EXTERNAL PORT

Parameter CASC=0, EBUS=1



CONNECTION OF INTERNAL & EXTERNAL CORES TO CPU

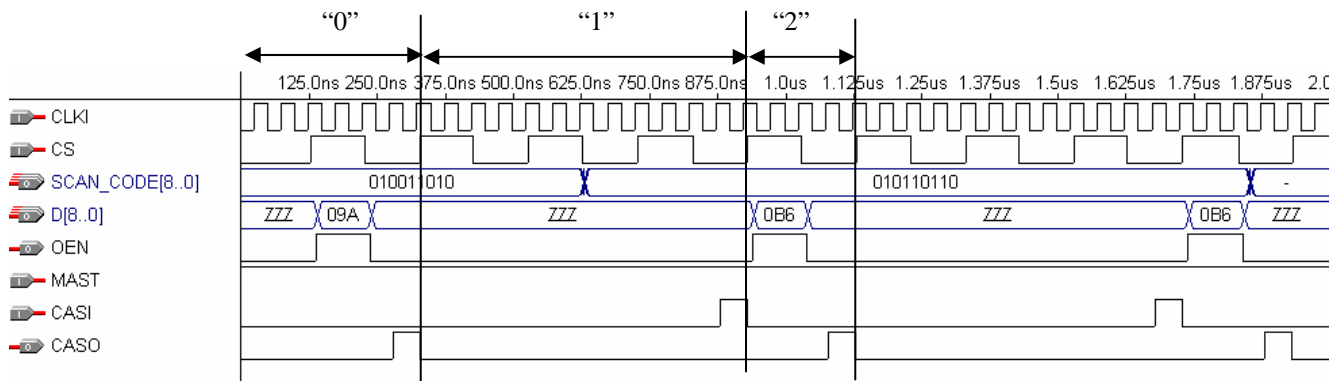




UNREGISTERED SCAN CODE-CASCADED - Parameters CASC=1,IREG= 0

- 1)Read scan code - (section “0” in timing diagram)
- 2)Read data from cascaded peripheral - (section “1” in timing diagram)
- 3)Read scan code - (section “2” in timing diagram)

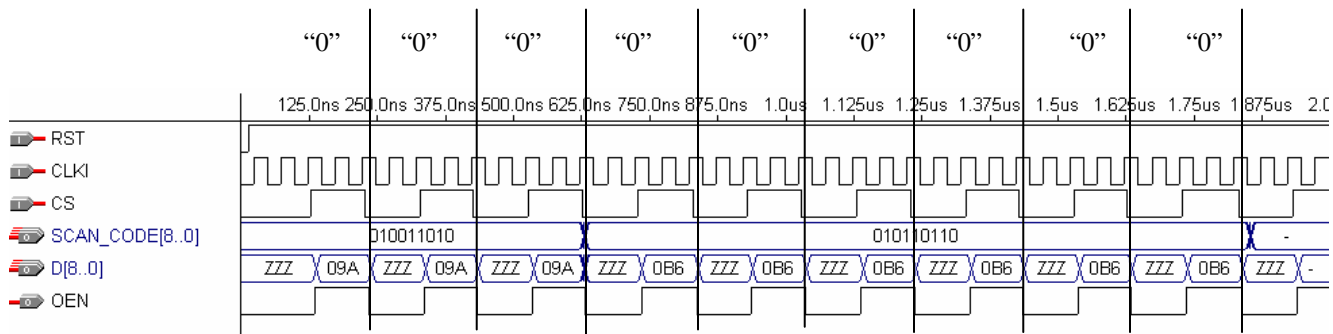
Parameter settings: IREG=0,OREG=1,KBD=1,NMR=16,NMC=16,DCN=0,EBUS=1/0,DBW=9,CASC=1,CEF=0, GCL=20000000



UNREGISTERED SCAN CODE-STAND-ALONE - Parameters CASC=0,IREG= 0

- 1)Read scan code - (section “0” in timing diagram)

Parameter settings: IREG=0,OREG=1,KBD=1,NMR=16,NMC=16,DCN=0,EBUS=1/0,DBW=9,CASC=0,CEF=0, GCL=20000000





PARAMETERS AND PORTS

INPUT PORTS

NAME	DESCRIPTION	WIDTH	COMMENTS
MAST	Master select	1	When the core is a master in a cascaded configuration(CASC=1), set MAST to Hi. In a cascaded configuration if it is not the master, set it Lo. In a stand-alone configuration(CASC=0), it is unused and may be left open.
CASI	Cascade in	1	If the core is in a cascaded configuration(CASC=1) and is not the master, connect CASI to CASO from the previous core in the chain, if it is the master, set it to CASO of the last core in the chain. In a stand-alone configuration(CASC=0) it is unused and may be left open.
CLKI	Clock	1	Positive edge triggered. Synchronizes all internal operations
RST	Reset	1	Asynchronous, active lo, resets all internal logic
CS	Chip select	1	Active hi, enables the internal data read logic. Must be synchronous to the rising or falling edge of the CLKI input. The TRI-STATE buffers are enabled while CS is hi and scan-code data appears on the 'D' port. See DBIR.doc, w/ CLTL=0 for CS timing constraints.
ENB	Clock enable	1	Periodic pulses of 1 CLKI width and frequency of CEF. When unused connect to '1'
ROW	Keyboard row	NMR	Used only when OPN=0, else may be connected to any constant value

OUTPUT PORTS

NAME	DESCRIPTION	WIDTH	COMMENTS
D	Data bus	DBW	TRI-STATE buffers are enabled and scan code appears on it while CS is hi
COD	Column	NMC	Time division multiplexed (TDM) keyboard column output signals
CASO	Cascade out	1	Used in a cascaded configuration (CASC=1), to enable the next core in the chain for data loading from the data bus. In a stand-alone configuration, it is unused and drives out Lo.
CEN	Scan frequency	1	5000 KHz keyboard scan clock w/ Hi period of 1CLKI pulse, for designer to use as needed

PARAMETERS-all INTEGER type

NAME	MIN	DESCRIPTION
IREG	0	Scan-code is latched after keypress until read -Yes (1) , No (0) .
OREG	0	COD[]output is registered (1) , combinatorial (0)
KBD	1	Reserved. Leave 1 or unused
NMR	1	Number of rows
NMC	1	Number of columns
DCN	0	Debounce count for ROW[] inputs (0-for no debounce)
EBUS	0	Scan code (D[]) is taken out of the chip, onto PCB – Yes (1), No (0)
DBW	1	Width of D[] output port. See “INTERFACE INFORMATION” above
CASC	0	Core is Cascaded/Standalone –1/0. Determines the behaviour of MAST and CASI inputs and the CASO output.
GCL	0	Frequency of CLKI input. GCL>=5 KHz
CEF	0	Frequency of ENB input. CEF=0 if ENB='1', else CEF>=5 KHz