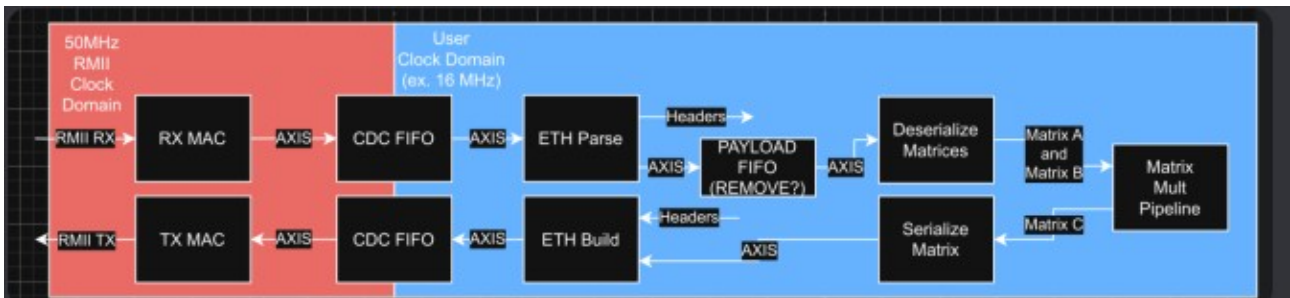


\*\*\*\*\*Default\*\*\*\*\*

## PipelineC PMOD Ethernet connected to a pico2-ice 04/28/25

\*\*\*\*\*Default\*\*\*\*\*

The current image which does work in the form matrix multiply.



with those and the work() pipeline looks like so



Can come over to [# pipelinec](#) to hear more about this first demo of configuring the inputs work func and outputs to do 2x2 int8 matrix multiplication

Tested 03/06/25 in pico-ice repo branch test-dev-100424 in two shell below.

Following “git clone <https://github.com/develone/pico-ice.git> -b test-dev-100424”

If the command in the left image below is done before the command in right image below.

```
devel@pi5-70:~/pico-ice/PipelineC/pmod-ethernet/examples/net $ ./compile-work_test.sh
```

The work\_test.c is dependent on the successful build of gateway.bin see Appendix A.

In file included from work\_test.c:30:

```
work.h:26:10: fatal error: type_bytes_t.h/work_inputs_t_bytes_t.h/work_inputs_t_bytes.h: No such file or directory
```

```
26 | #include "type_bytes_t.h/work_inputs_t_bytes_t.h/work_inputs_t_bytes.h"
```

```
| ^~~~~~
compilation terminated.
```

```
make clean all TOP_NAME=ethernet_top NEXTPNR_ARGS="--pre-pack eth_clocks.py"
```

```
rm -f lextab.py
```

```
rm -f yaccstab.py
```

```
rm -f pll_clk_mhz.h
```

```
rm -f -r pipelinec_output
```

```
rm -f pipelinec.log
rm -f pll.v
rm -f *.json *.asc *.bin *.uf2
rm -f yosys_stderr.log
rm -f dfu_util.log
echo "#define PLL_CLK_MHZ 25.0\n" > pll_clk_mhz.h
/home/devel/PipelineC/src/pipelinec ethernet_top.c --top pipelinec_top --out_dir
pipelinec_output --comb --no_synth > pipelinec.log
/home/devel/oss-cad-suite/bin/icepll -q -i 12 -o 25.0 -p -m -f pll.v
/home/devel/oss-cad-suite/bin/yosys -l simple.log -q -m ghdl -p "ghdl --std=08 -frelaxed `cat
pipelinec_output/vhdl_files.txt` -e pipelinec_top; read_verilog -sv ethernet_top.sv pll.v;
synth_ice40 -top ethernet_top -json gateway.json" 2> yosys_stderr.log
/home/devel/oss-cad-suite/bin/nextpnr-ice40 -l nextpnr.log -q --randomize-seed --up5k --
package sg48 --pcf ice40.pcf --json gateway.json --asc gateway.asc --pre-pack eth_clocks.py
/home/devel/oss-cad-suite/bin/icepack gateway.asc gateway.bin
```

```
devel@pi5-70:~/pico-ice/PipelineC/pmod-ethernet/examples/pico/ice_makefile_pipelinec $
bin2uf2 -o gateway.uf2 gateway.bin
```

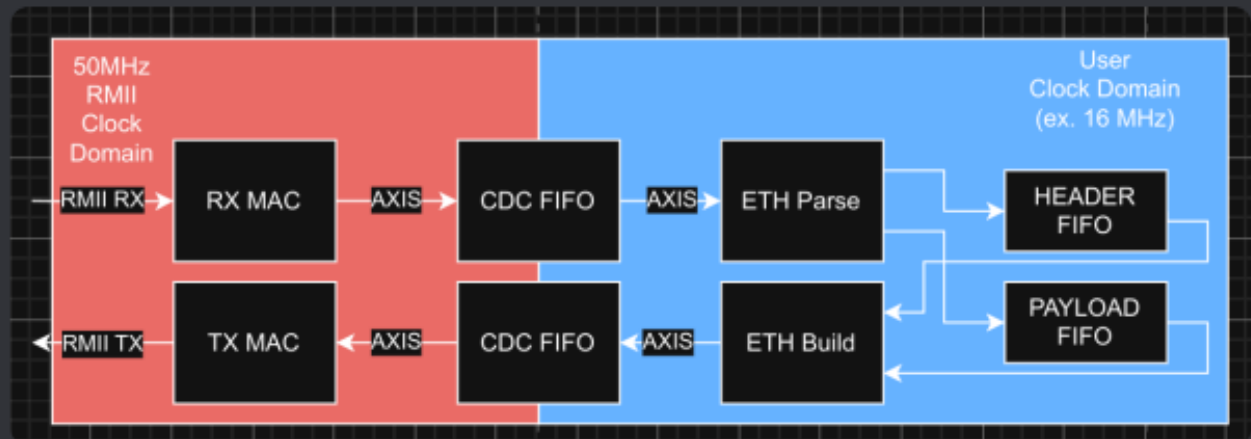
```
devel@pi5-70:~/pico-ice/PipelineC/pmod-ethernet/examples/pico/ice_makefile_pipelinec $ scp
gateway.uf2 pi5-90:/media/devel/pico-ice/
gateway.uf2                               100% 204KB 10.4MB/s 00:00
```

```
devel@pi5-70:~/pico-ice/PipelineC/pmod-ethernet/examples/net $ ./compile-work_test.sh
```

```
devel@pi5-70:~/pico-ice/PipelineC/pmod-ethernet/examples/net $ sudo ./work_test
CPU threads: 1
n 'work()'s: 1
Total tx bytes: 8
Total rx bytes: 4
CPU took 0.000290 seconds to execute
CPU iteration time: 0.000290 seconds
CPU bytes per sec: 41391.157895 B/s
FPGA took 0.000207 seconds to execute
FPGA iteration time: 0.000207 seconds
FPGA bytes per sec: 57985.769585 B/s
Speedup: 1.400922
```

This image was loopback tested on 02/17/25

this is what the current design implements



AXIS buses are 8b wide

user clock could be as low as  $50M/4 = 12.5M$

pico-ice Running gateway.bin provided by Discord user absurdfatalism. Led blink red.

This was fixed with added make parameters.

make clean

make pipelinec TOP\_NAME=ethernet\_top NEXTPNR\_ARGS="--pre-pack eth\_clocks.py"

make gateway.bin TOP\_NAME=ethernet\_top NEXTPNR\_ARGS="--pre-pack eth\_clocks.py"

bin2uf2 -o gateway.uf2 gateway.bin

scp gateway.uf2 pi5-90:/media/devel/pico-ice/

XX

```
devel@pi5-90: ~
File Edit Tabs Help
Welcome to minicom 2.8

OPTIONS: I18n
Port /dev/ttyACM0, 11:54:40

Press CTRL-A Z for help on special keys

tud_cdc_line_coding_cb: coding=20001278 baud=115200

pico-ice default firmware
https://github.com/tinyvision-ai-inc/pico-ice/tree/main/Firmware/pico-ice-det

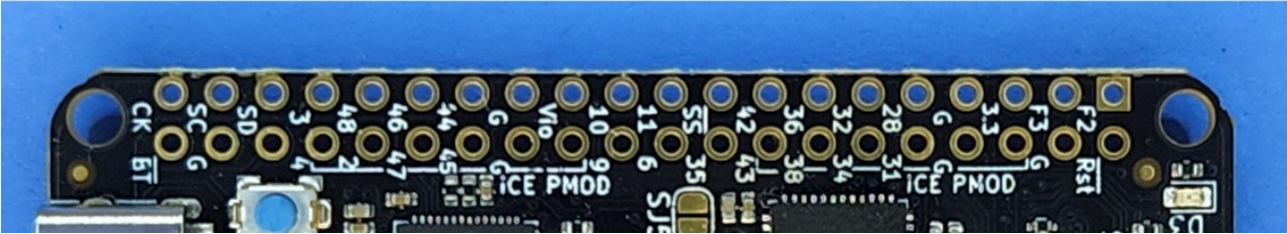
Serial port #0 - this shell, with commands:
v - print pico-ice-sdk version

Serial port #1 - forwarding to UART
UART TX on RP20 = ICE27
UART RX on RP30 = ICE25

Serial port #2 - forwarding to SPI:
https://pico-ice.tinyvision.ai/group__ice__usb.html#autotoc_md2

pico2-ice> default Firmware to provide a 12 MHz clock
```

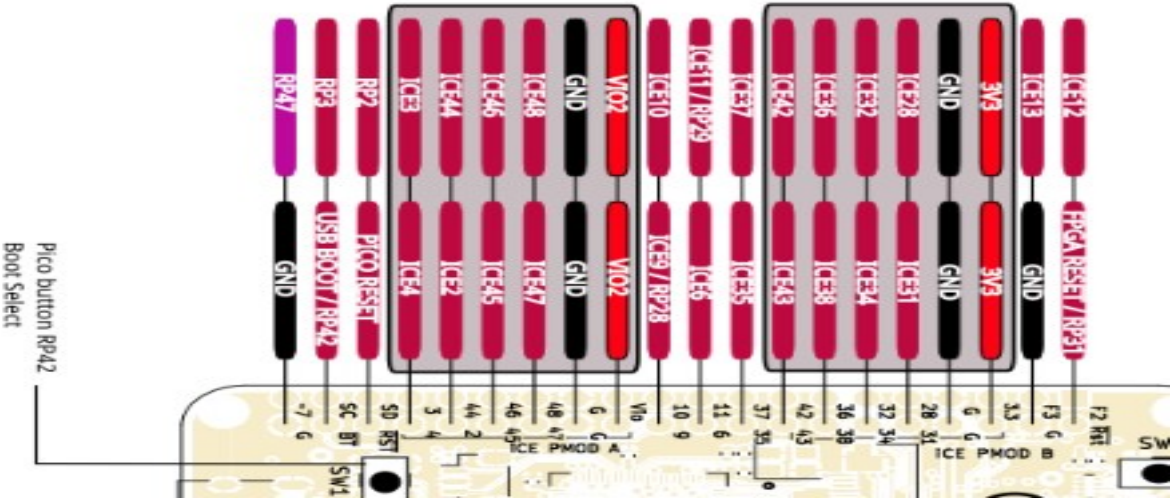
XX



outer row bottom 3 48 46 44 42 36 32 34

inner row top 4 2 47 45 43 38 34 31

pico-ice

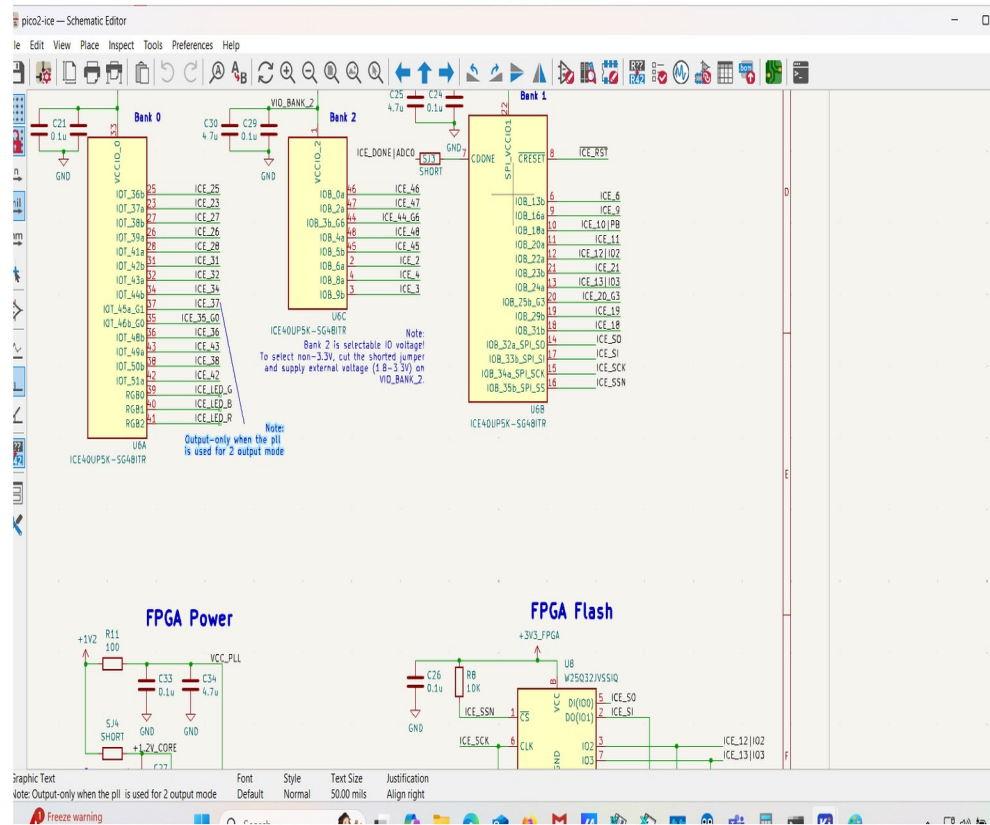


outer row bottom 3 44 46 48 42 36 32 34

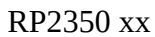
inner row top 4 2 45 47 43 38 34 31

pico2-ice

XX







3.3 grd A4 A3 A2 A1



3.3 grd B4 B3 B2 B1

A1-43 B1-42  
A2-38 B2-36  
A3-34 B3-32  
A4-31 B4-28

pico-ice

3.3 grd A4 A3 A2 A1

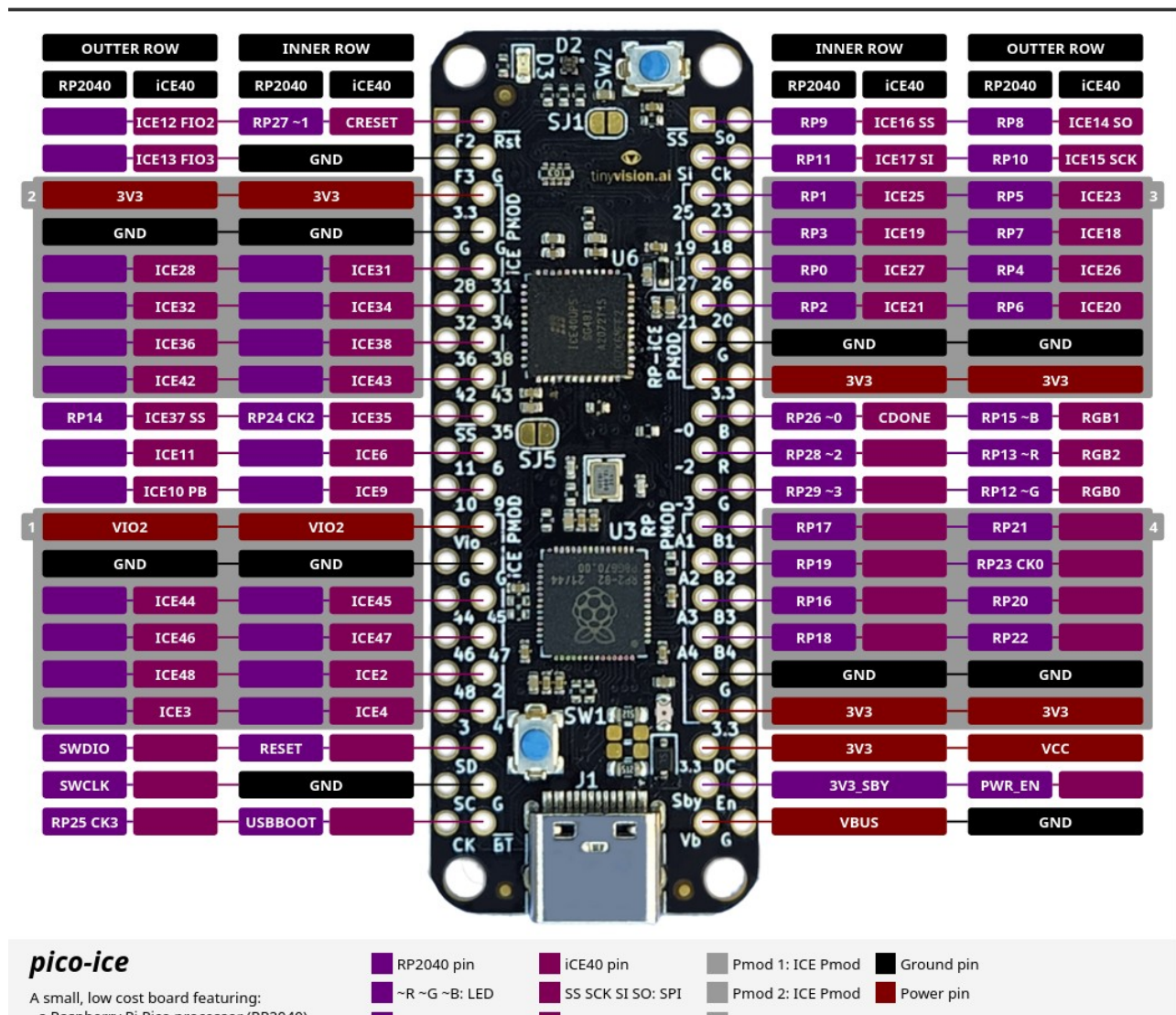


3.3 grd B4 B3 B2 B1

A1-4 B1-3  
A2-2 B2-8  
A3-47 B3-46  
A4- 45 B4-44

USB

XX





```
pi5-70 or pi5-80
devel@pi5-80:~/pico-ice/PipelineC/pmod-ethernet/ice_makefile_pipelinec $
devel@pi5-70:~/pico-ice/PipelineC/pmod-ethernet/ice_makefile_pipelinec $
make clean
make pipelinec TOP_NAME=ethernet_top NEXTPNR_ARGS="--pre-pack eth_clocks.py"
make gateway.bin TOP_NAME=ethernet_top NEXTPNR_ARGS="--pre-pack eth_clocks.py"
bin2uf2 -o gateway.uf2 gateway.bin
scp gateway.uf2 pi5-90:/media/devel/pico-ice/
devel@pi5-90:~/PipelineC/examples/arty/src/eth $ sudo ./loopback_test
Test passed!
```



## Appendix A

```
#pragma once
#include "type_bytes_t.h/int8_t_bytes_t.h/int8_t_bytes.h"
#define work_inputs_t_SIZE 8

void work_inputs_t_to_bytes(work_inputs_t* x, uint8_t* bytes)
{
    size_t pos = 0;
    // matrix0
    size_t matrix0_dim_0;
    for(matrix0_dim_0=0;matrix0_dim_0<2;matrix0_dim_0=matrix0_dim_0+1){
        size_t matrix0_dim_1;
        for(matrix0_dim_1=0;matrix0_dim_1<2;matrix0_dim_1=matrix0_dim_1+1){
            int8_t_to_bytes(&(x->matrix0[matrix0_dim_0][matrix0_dim_1]), &(bytes[pos]));
            pos = pos + 1; // not sizeof()
        }
    }
    // matrix1
    size_t matrix1_dim_0;
    for(matrix1_dim_0=0;matrix1_dim_0<2;matrix1_dim_0=matrix1_dim_0+1){
        size_t matrix1_dim_1;
        for(matrix1_dim_1=0;matrix1_dim_1<2;matrix1_dim_1=matrix1_dim_1+1){
            int8_t_to_bytes(&(x->matrix1[matrix1_dim_0][matrix1_dim_1]), &(bytes[pos]));
            pos = pos + 1; // not sizeof()
        }
    }

}

void bytes_to_work_inputs_t(uint8_t* bytes, work_inputs_t* x)
{
    size_t pos = 0;
    // matrix0
    size_t matrix0_dim_0;
    for(matrix0_dim_0=0;matrix0_dim_0<2;matrix0_dim_0=matrix0_dim_0+1){
        size_t matrix0_dim_1;
        for(matrix0_dim_1=0;matrix0_dim_1<2;matrix0_dim_1=matrix0_dim_1+1){
            bytes_to_int8_t(&(bytes[pos]), &(x->matrix0[matrix0_dim_0][matrix0_dim_1]));
            pos = pos + 1; // not sizeof()
        }
    }
    // matrix1
    size_t matrix1_dim_0;
    for(matrix1_dim_0=0;matrix1_dim_0<2;matrix1_dim_0=matrix1_dim_0+1){
        size_t matrix1_dim_1;
        for(matrix1_dim_1=0;matrix1_dim_1<2;matrix1_dim_1=matrix1_dim_1+1){
            bytes_to_int8_t(&(bytes[pos]), &(x->matrix1[matrix1_dim_0][matrix1_dim_1]));
            pos = pos + 1; // not sizeof()
        }
    }
}
```

}