

XMOS の XS1-G4,L1 であそぶ - その 3

テスト基板 XC-1,XC-2,XK-1 などであそんで気がついたことのメモを書いていきます。

Stack available: FAILED って ? (9.9.0)

ちょっと不親切なメッセージですが、「メモリが足りない時」に出ます。コードもデータもコアあたり 64KB 共有なので、メモリがスタック領域まで行くと、

```
Constraint check for node 0, core 0:
Stack available: <00000000, used: 00000008 . FAILED
Threads available:      8, used:      3 . OKAY
Timers available:     10, used:      2 . OKAY
Chanends available:   32, used:      0 . OKAY
```

のようにおこられます。
メモリを使いすぎているか、見直しましょう。

chan と streaming chan (9.9.0 & XS1-L1)

chan は完全同期で、受け取り側が入力待ちでないとブロックします。一方、streaming chan はバッファを持っており、非同期でデータを受け渡せます。

引き取り側がないときのバッファに先行して押し込めるデータ数は、下記の実験によると、2ワード(8バイト)までです。

実験プログラム

```
#include <xs1.h>
#include <platform.h>
#include <print.h>

void produce(streaming chanend c)
{
    int i=0 ;
    while(1) {
        c <: i ;
        printintln(i) ;
        i++ ;
    }
}

void consume(streaming chanend c)
{
    int i=0 ;
    while(1){
        i++ ;
    }
}

int main(void)
{
    streaming chan c ;
```

```

    par {
        produce(c) ;
        consume(c) ;
    }
    return 0;
}

```

このプログラムの出力

```

0
1

```

で停止します。

produce の c にデータを喰わせる箇所を

```

c <: (unsigned char) i ;
2013/1 (unsigned) から (unsigned char) に訂正

```

とすると、0...7 で停止します。

余談ですが、streaming channel で 32bit を下回るデータを取り扱う表現を select 内で使うと、Exception で落ちることがあります。

現バージョン(9.9.0)の開発環境が吐くコードのせいかと思いがちですが、どうも仕様のように、資料にそれらしき記述があります。ただし、こちらには "in the guard of a select statement." と限定的に書いてありますので、私が経験した Exception とは違う可能性もあります。

Programming XC on XMOS Devices (2009/09/24 版) から引用

C.4 Channel Communication

On some revisions of the XS1 architecture, it is not possible to input data of size less than 32 bits from a streaming channel in the guard of a select statement.

32bit 以下のデータをやりとりしていて、どうもうまくいかなかったら、デバッガでその箇所でもまわっていないか、32bit 型にするとうまく動いたりしないか、試してみるといいと思います。

streaming chan への出力がブロックするかどうかを知る (9.9.0)

チャンネルからの入力待ちは select で行えますが、出力してもブロックしないかどうかは出力方向には select が使えないので、わかりません。

逆方向の通信が空いているときは、下記の例のように、受ける側であらかじめ入力できることを通知することで、出力可能検査を入力待ちの select に変換することができます。

[参考] <http://www.xmoslinkers.org/forum/viewtopic.php?f=6&t=617>

```

void consume( streaming chanend c )
{
    unsigned v;
    const unsigned ready = 0xff;

```

```

while (1)
{
    c <: ready; //ready to receive more
    c >: v; // get data now
    delay(100); // sample processing delay
}
}

void produce( streaming chanend c )
{
    unsigned v = 100;

    while(1)
    select
    {
        case c >: int:
            c <: v;
            printstrln("Send");
            break;
        default:
            /* put into buffer */
            printstrln("buffer");
            break;
    }
}
}

```

xflash がエラーになることがある (9.9.1)

「その2」で解説した、2つのXS1-L1(XK1)を接続する.XNファイルを含むコードが、9.9.1に含まれるxflashのエラーでSPIメモリに焼けないことがあるようです。

具体的には以下のようなエラーで止まってしまいます。

```

portDeclsXC_a04656:32: error: use of `deviceSpecs' violates parallel usage rules
portDeclsXC_a04656:28: error: previously used here (bytes 0..420)
portDeclsXC_a04656:28: error: previously used here (bytes 0..420)
Error: F03010 Failed to compile flasher app.

```

XMOSのサポートの方にエラー報告済みです(09/12/7)。

確認していただいたようですので、次のバージョンでは改善されているかと思えます(・・・)

XS1-G と L の違い：I/Oの内蔵抵抗について

いままで気にしなかったのですが、ふとスイッチを外付けしたときに、横着してGと同様に内蔵プルアップ(ライブラリにポートに対する操作が存在する)を使おうとして、初めて気づきました。

ライブラリリファレンスのset_port_pull_up()の項より引用

```

On XS1-L devices no pull-up resistors are available
and an exception will be raised if
set_port_pull_up() is called.

```

逆に、LにはGには存在しない内蔵プルダウンがあるようです。ライブラリリファレンスのset_port_pull_down()の項より引用

```

On XS1-G devices no pull-down resistors are
available and an exception will be raised if
set_port_pull_down() is called.

```

注意すべき点は、L で pull_up をしようとしたり、G で pull_down をしようすると Exception が発生 = プログラムが停止します。これはデバッガで動かしていないとなかなか気づかないので、要注意です。

<< XMOS の XS1-G4 であそぶ - その 2 | XMOS の XS1-G4.L1 であそぶ - その 4 >>