

Artix-7のDDR機能活用 (中性子検出器回路への応用)

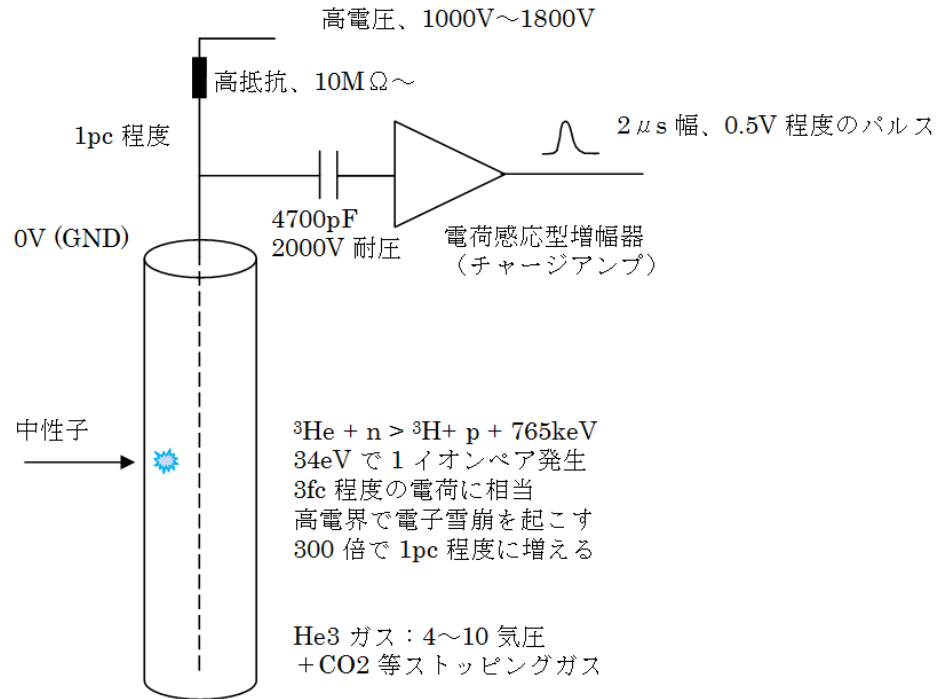
高エネルギー加速器研究機構
中性子研究系・佐藤節夫

発表内容

中性子検出器読み出し回路のDDR活用
Spartan6から、Artix7へ、DDR機能の変化
8入力ADC-ICのDDR読出し
SDRAMのDDRデータアクセス
モジュール間のDDRデータ送受信
Vivadoのデバッグ機能
まとめ

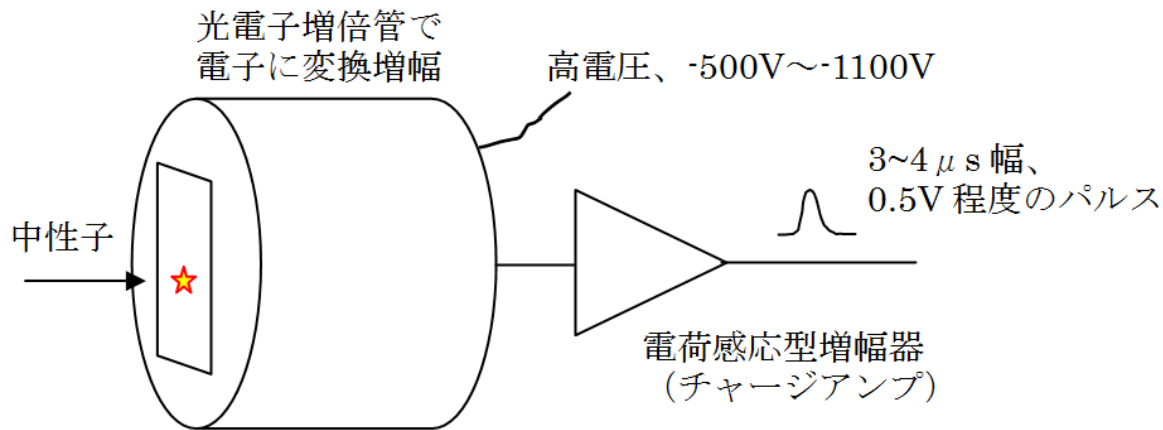
中性子検出の王道、 ^3He ガス検出器

中性子と強く
反応する ^3He
ガスを使用す
る。構造が簡
単で圧力が上
げられるので、
高検出効率に
できる。



中性子が ^3He と核反応を起こし、発生エネルギーで数万個の電子が発生し、ガス増幅で1pC程度になる。

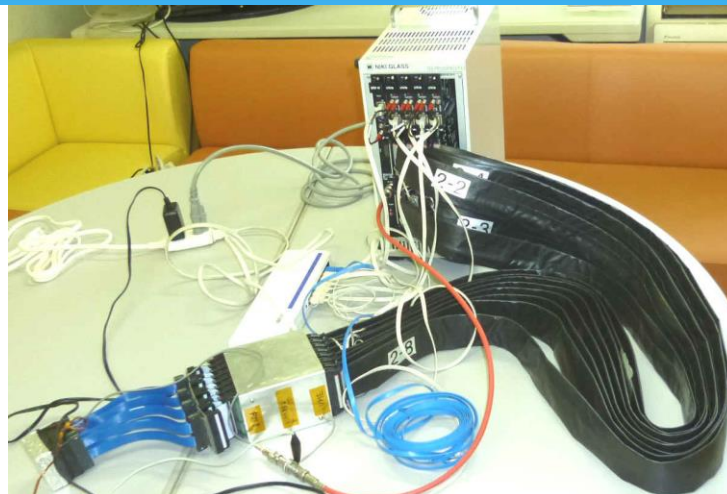
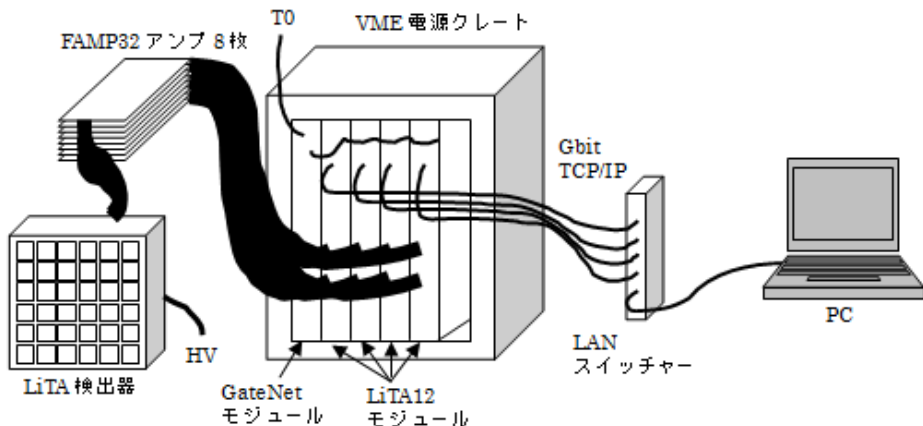
中性子で光る： ${}^6\text{Li}/\text{ZnS}$ シンチレータ



${}^6\text{Li}/\text{ZnS}$ シンチレータ
 ${}^6\text{Li} + n > {}^3\text{H} + {}^4\text{He} + 4.8\text{MeV}$
周りの ZnS で約 16 万光子が発生

中性子が ${}^6\text{Li}$ と核反応を起こし、4.8MeVを発生し、運動エネルギーとなってトリチウムとヘリウムがはじき出される。周りのZnSシンチレータが励起され、光子が発生する。>>微小な光なので、**光電子増倍管**(photo-multiplier tube)で電子に変換し、100万倍ぐらいに増幅する。

DDRを多用する、LiTA12中性子検出器



LiTA12システムの構成図

検出器領域: 49mm x 49mm

<高計数率モード>

位置分解能: 3mm

最大計数率: 50Mcps

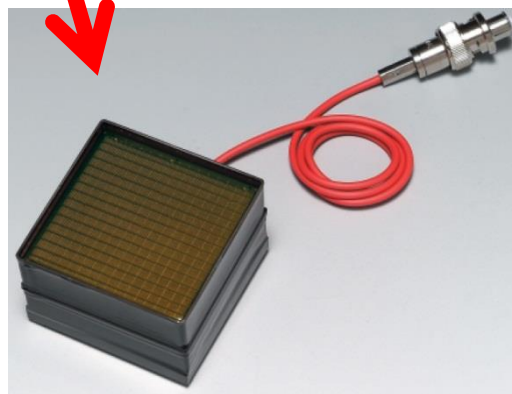
検出効率: 約40%

<高位置分解能モード>

位置分解能: 0.4mm

最大計数率: 5Mcps

検出効率: 使用するシンチレータによる

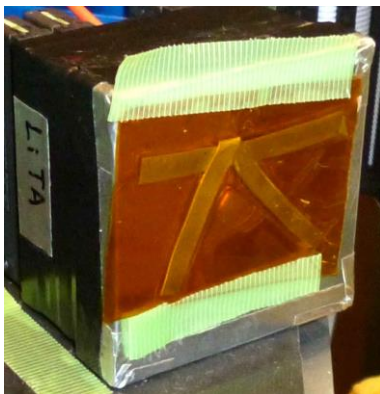


浜ホトH9500、16x16のマルチアノード光電子増倍管(PMT)を使用。

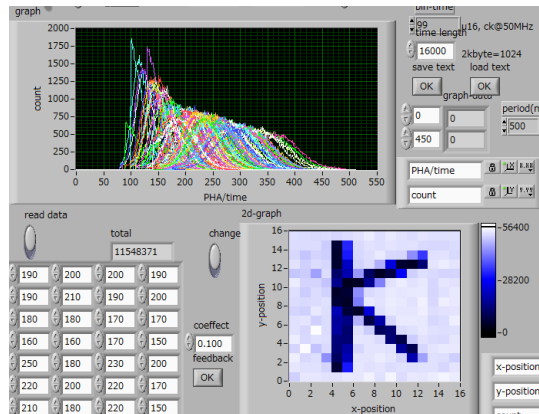
高計数率測定



16×16に独立にシンチレータを配置し、クロストークさせない

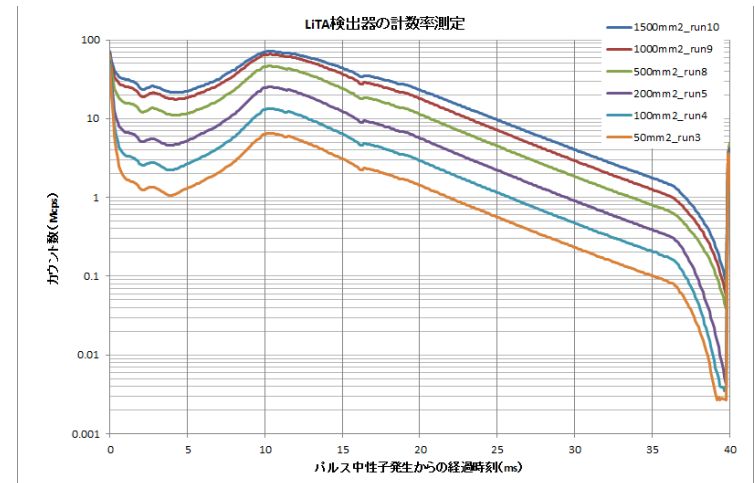


カドミウム「K」文字



測定データ

照射面積—時間分布図から、50Mcps得られた。

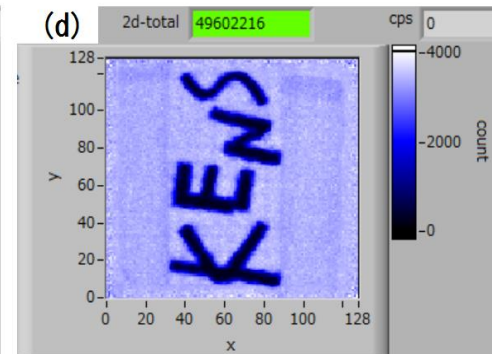
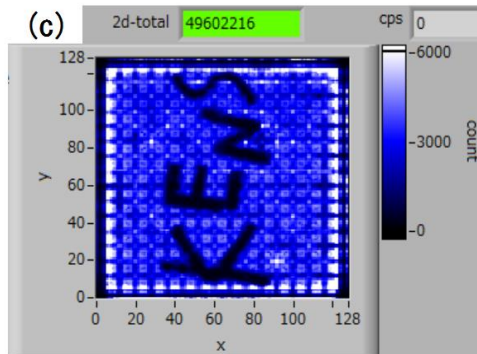
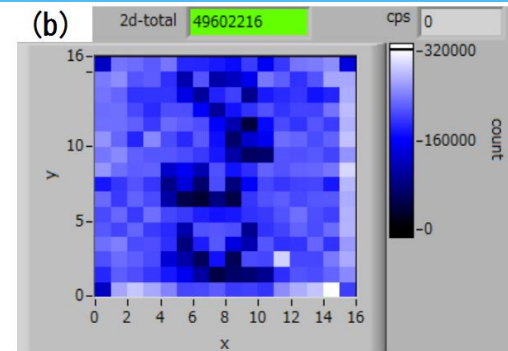


照射面積—時間分布特性

高位置分解能測定



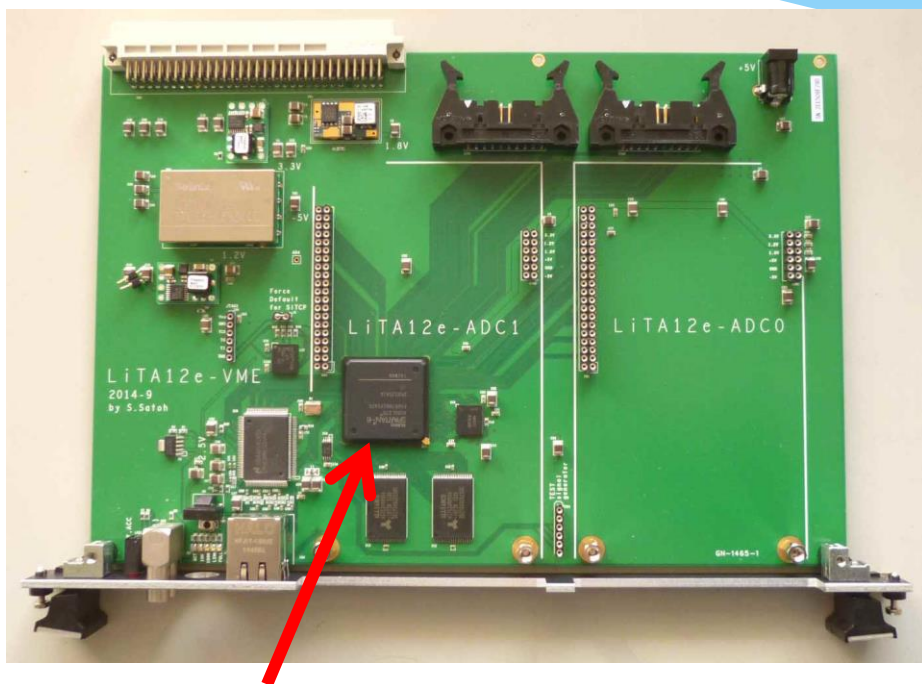
1枚のシンチレータでクロストークさせている>>重心計算できる



(a) Cd文字、(b) 3mmピクセル、(c) 0.4mmピクセル、(d) 0.4mmピクセル

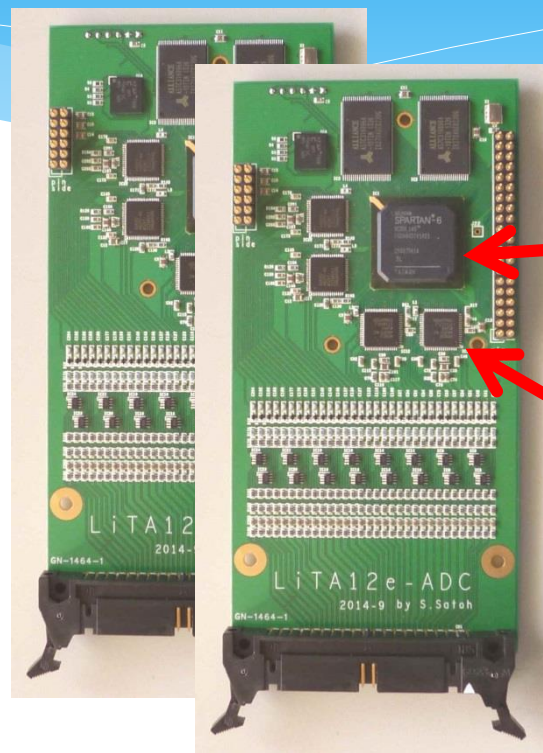
強度補正でテープ跡(水素)が見える

VMEモジュールとADC子基板



比較的大きなFPGA

LiTA12-VMEモジュール



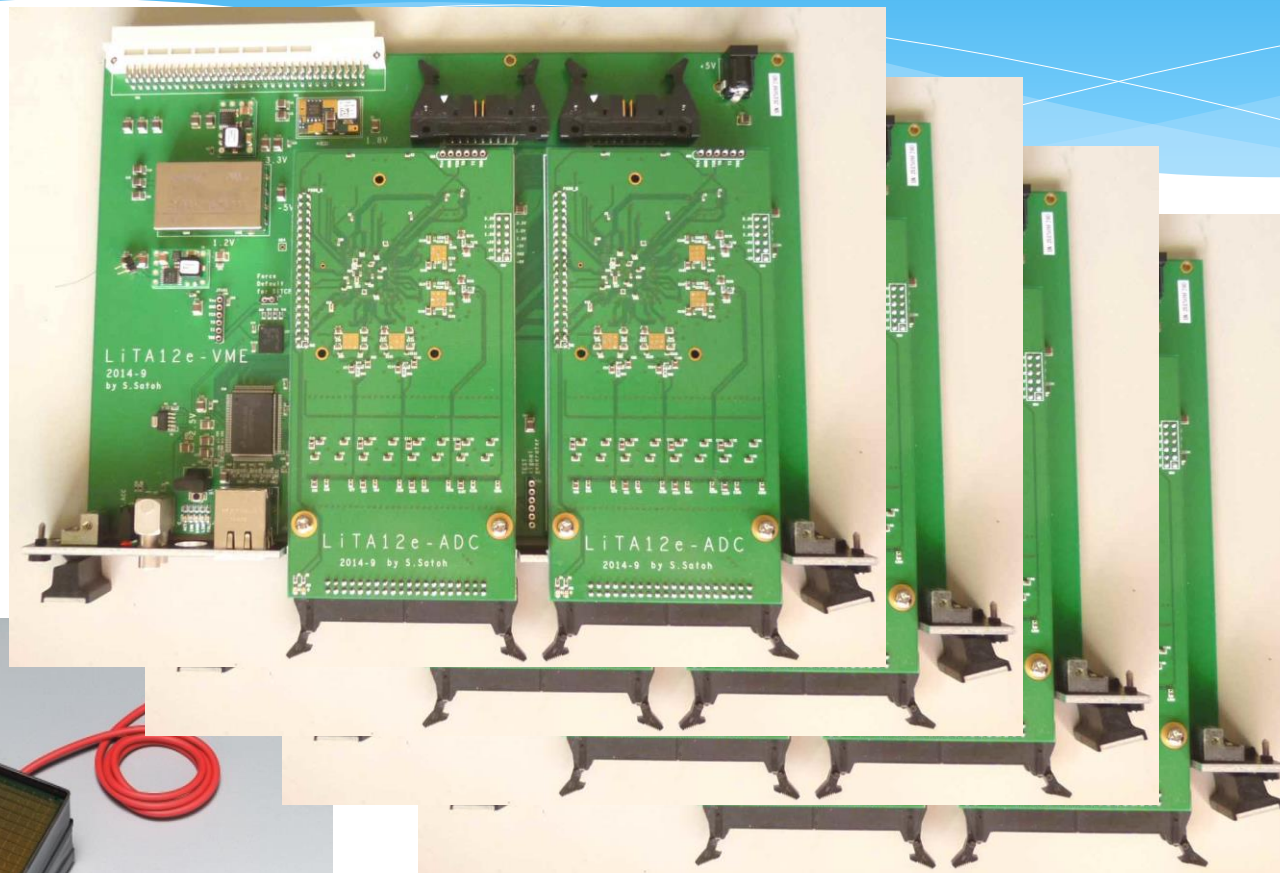
ADC用
FPGA

ADC
8入力
× 4個

LiTA12-ADC基板: 2枚載る
>> 32ピクセル × 2

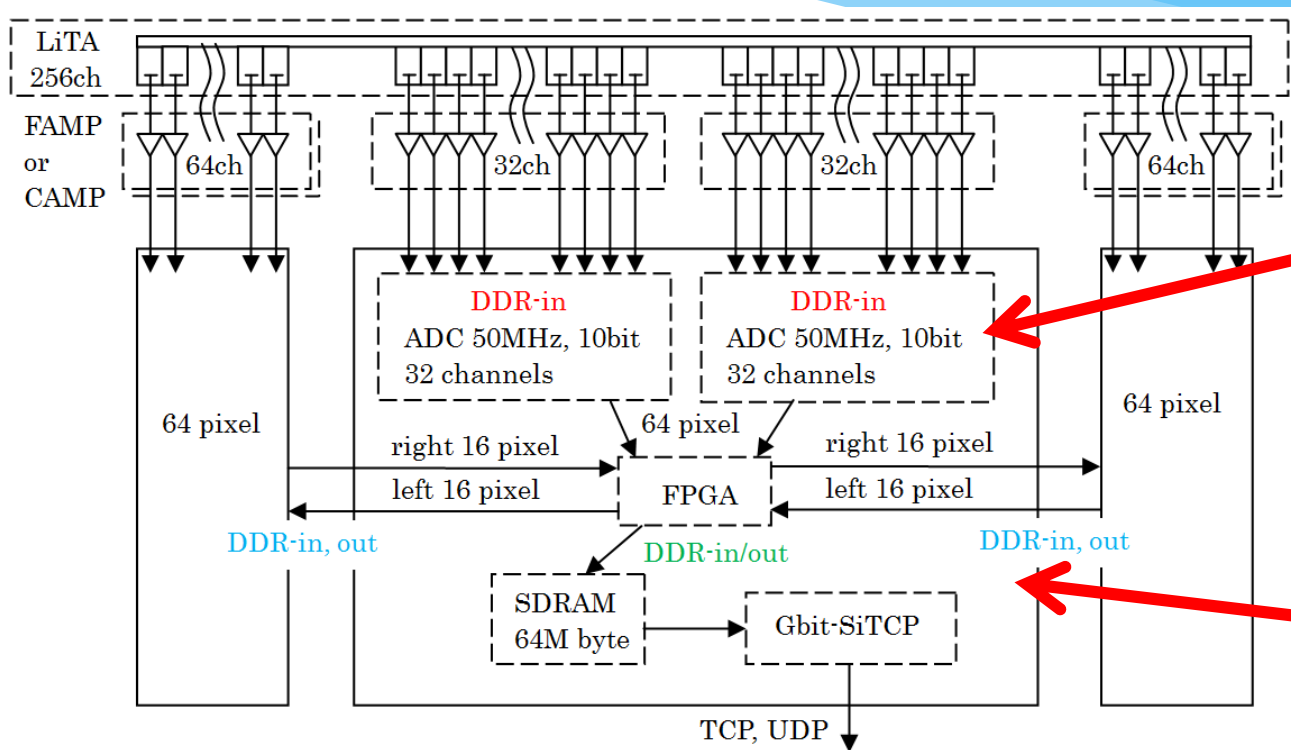
1枚のVMEモジュールに64chのADCを載せるため、子基板で高密度化

VMEモジュールとADC子基板



16 × 16-PMT = 64ピクセルVMEモジュール × 4台
>> 256ピクセルを処理できる。

LiTA12中性子検出器のDDR機能



DDR制御部分:ADC読出し(赤)、SDRAM制御(緑)、モジュール間データ送受信(青)。

LiTA12検出器で、DDRを多用

- ◆ ADC読出し：**8入力ADCを8個**使用。>> 500Mbpsのため、DDR受信。
- ◆ **SDRAM(64MB)** :SiTCP用のバッファメモリに使用している。**独自開発のDDR**送受信。
- ◆ **モジュール間のデータ送受信** :隣のADC出力をDDR送受信。

8入力ADCのDDR入力

8入力ADC: 8個/モジュール



Octal, 12-Bit, 40/80 MSPS, Serial LVDS, 1.8 V Analog-to-Digital Converter

Data Sheet

AD9637

FEATURES

- Low power: 60 mW per channel at 80 MSPS with scalable power options
- SNR = 71.5 dBFS (to Nyquist)
- SFDR = 92 dBc (to Nyquist)
- DNL = ± 0.4 LSB (typical), INL = ± 0.5 LSB (typical)
- Serial LVDS (ANSI-644, default)
- Low power, reduced signal option (similar to IEEE 1596.3)
- Data and frame clock outputs
- 650 MHz full power analog bandwidth
- 2 V p-p differential input voltage range
- 1.8 V supply operation
- Serial port control
 - Full chip and individual channel power-down modes
 - Flexible bit orientation
 - Built-in and custom digital test pattern generation
 - Programmable clock and data alignment
 - Programmable output resolution
 - Standby mode
- APPLICATIONS
 - Medical imaging and nondestructive ultrasound
 - Portable ultrasound and digital beam-forming systems
 - Quadrature radio receivers
 - Diversity radio receivers
 - Optical networking
 - Test equipment

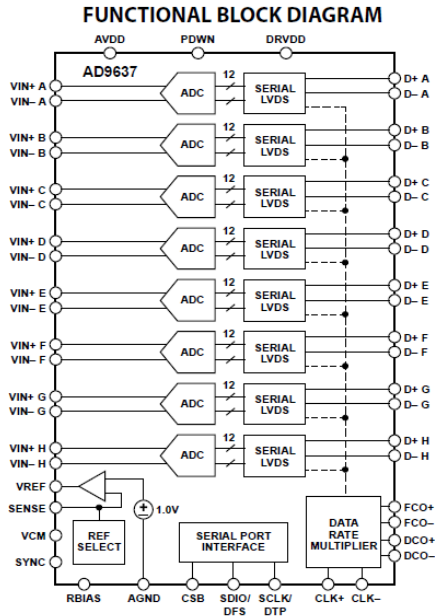


Figure 1.

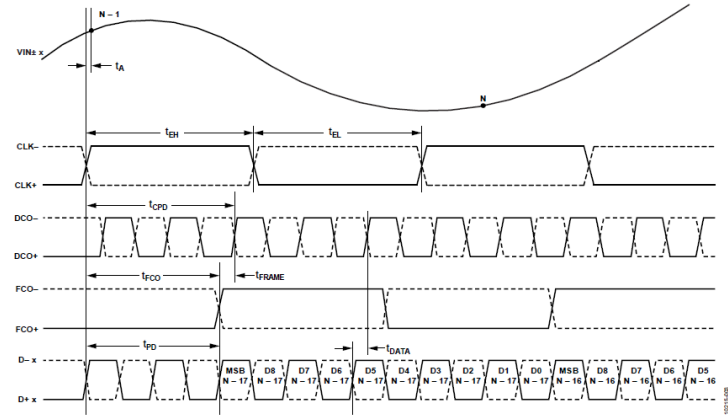
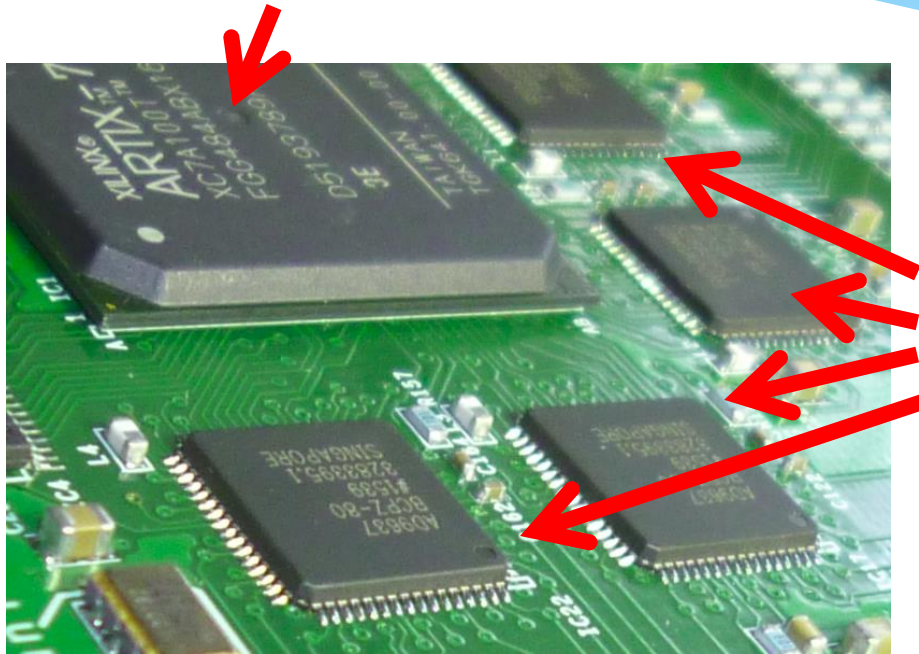


Figure 3. Word-Wise DDR, 1x Frame, 10-Bit Output Mode

50MHz-10bitをLVDS信号で受けるので、500Mbpsの転送速度が必要。差動LVDS信号で、DDRでクロックを250MHzに落とせるので可能になっている。

32入力ADCを500Mbpsで読み出す

Artix7, XC7A100T-FGG484



8入力ADC
× 4個

できるだけ近くに配置し、最短の配線を心掛けた。
50Ω、100Ω等の**終端抵抗が必要**であるが、Artix7ではいろいろ条件が厳しい。条件が悪く付いていなくてもエラーも出なかったなので、最近まで気が付かなかった。
ADCに付いていた終端抵抗をオンにして要件を満たした。

Artix7は7シリーズでは最低ランクなので、いろいろ条件が厳しいので注意が必要。

ADCをDDRで読み出す

Artix7での、高速シリアル出力ADCを ddrで読み出すプログラム

入力の遅延設定

constant IDELAY_VAL : integer:=5; --3,4,5,6 are ok, so 4 or 5 is best.

ID_1: **IDELAYE2** generic map(DELAY_SRC=>"IDATAIN", IDELAY_TYPE=>"FIXED", IDELAY_VALUE=>IDELAY_VAL)

port map(IDATAIN=>pfco, DATAOUT=>fco, C=>'0',

CE=>'0', CINVCTRL=>'0', CNTVALUEIN=>"00000", DATAIN=>'0', INC=>'0', LD=>'0', LDPIPEEN=>'0', REGRST=>'0');

u10: **IDDR** generic map (DDR_CLK_EDGE=>"SAME_EDGE_PIPELINED") port map(Q1=>fr0, Q2=>fr1, C=>pdckb, CE=>'1',D=>fco,
R=>'0', S=>'0');

DDR読み込み

IDDRに入れる前に、**信号の遅延を制御**できるようにし、波形を見ながら良い値を見つける。
波形が壊れる**最小値と最大値を求め、中間値に設定**する。

xdcファイルでの制約

クロックの相対値を指定した。

```
create_clock -name {adin[19]} -period 4.000 [get_ports {adin[19]}]
```

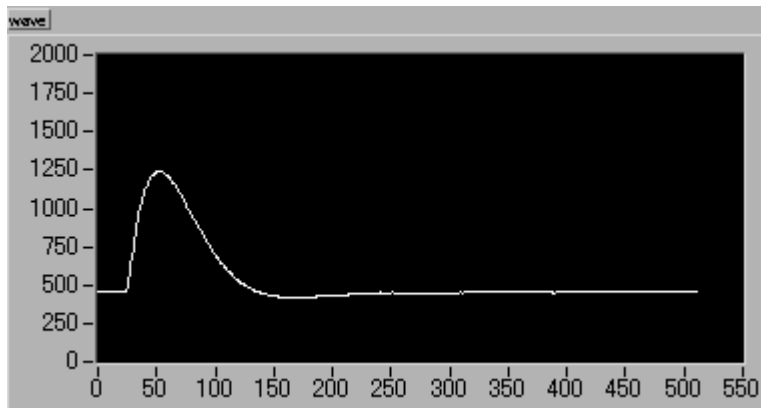
```
create_generated_clock -name {adin[18]} -source [get_ports {adin[19]}] -divide_by 1 -multiply_by 1 -invert [get_ports {da[18]}]
```

```
set_max_delay -from [get_clocks adin[18]] -to [get_clocks clk_out2_clk_wiz_0] 1.0 -datapath_only
```

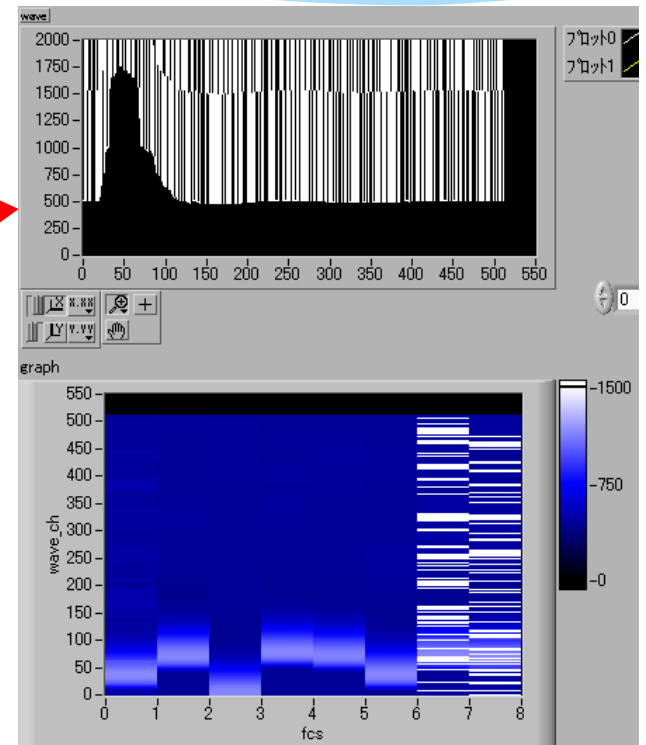
```
set_max_delay -from [get_clocks adin[19]] -to [get_clocks clk_out2_clk_wiz_0] 1.0 -datapath_only
```

ADCをDDRで読み出す

波形表示プログラムで確認



タイミングがずれると波形がデジタル的に崩れる。



IDDRに入れる前に、**信号の遅延を制御**できるようにし、波形を見ながら良い値を見つけろ。波形が壊れる**最小値と最大値を求め、中間値に設定**する。

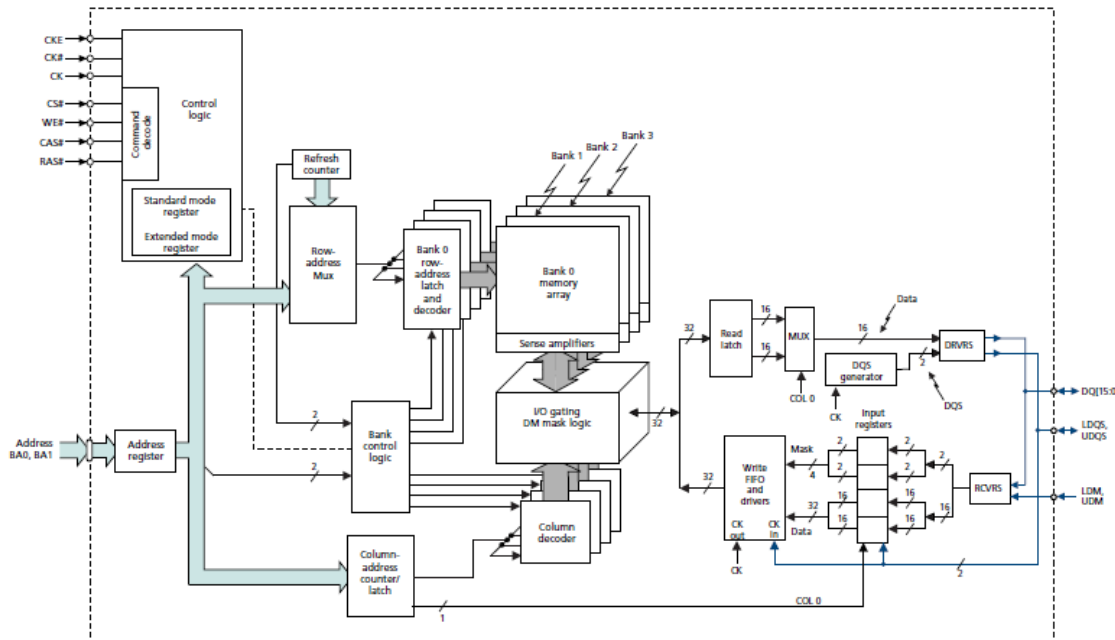
>>何回もコンパイルして確かめるので、時間がかかり、根気も必要。

右図はXC3S1400ANの遅延数－波形図。

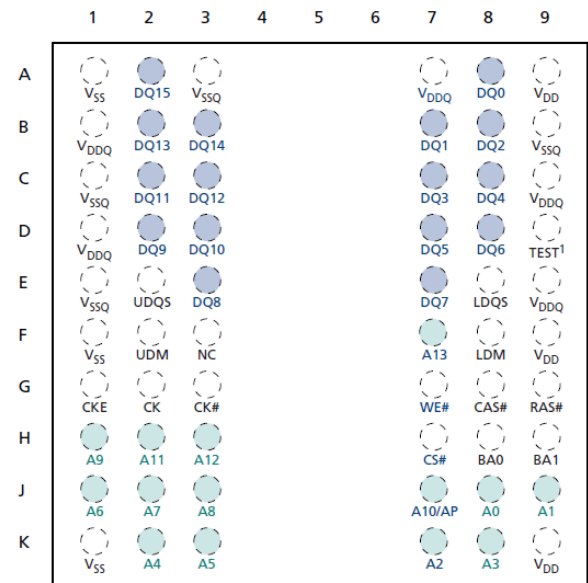
横軸:遅延数、縦軸:ADCクロック、濃淡:ADC値
中間の遅延数が安心。

SDRAMをDDRでアクセスする

SDRAM (MT46H32M16LF) ブロック図

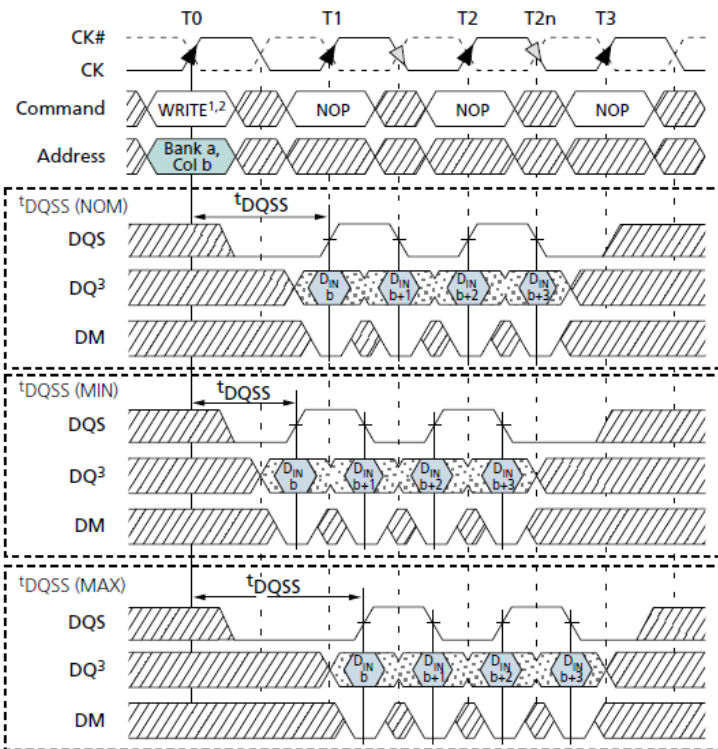


60-Ball VFBGA – Top View, x16 only



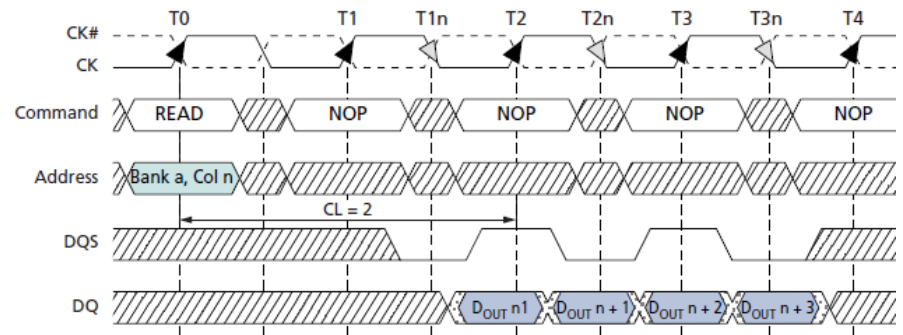
アドレスバスが2重構造になり、定期的なりフレッシュ、DDR操作等、使用方法が難しい。

SDRAMをDDRでアクセスする



書込み

書込みコマンド後、次のクロックから書き込む。
DQS信号で書き込みタイミングを決められる。



読出し

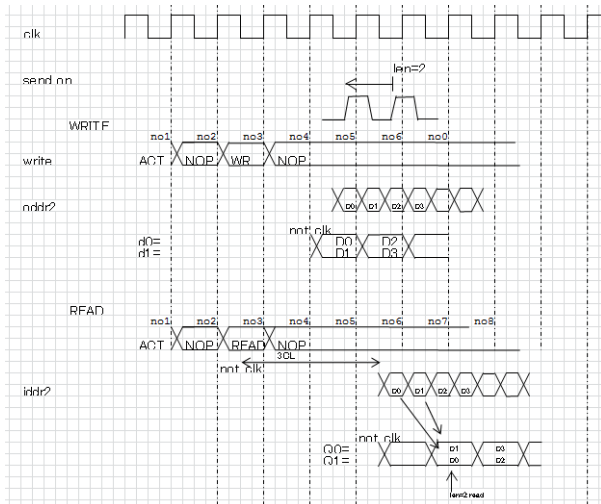
読出しコマンド後、レイテシークロック(CL)後から読み出せる。

SDRAMをDDRでアクセスする

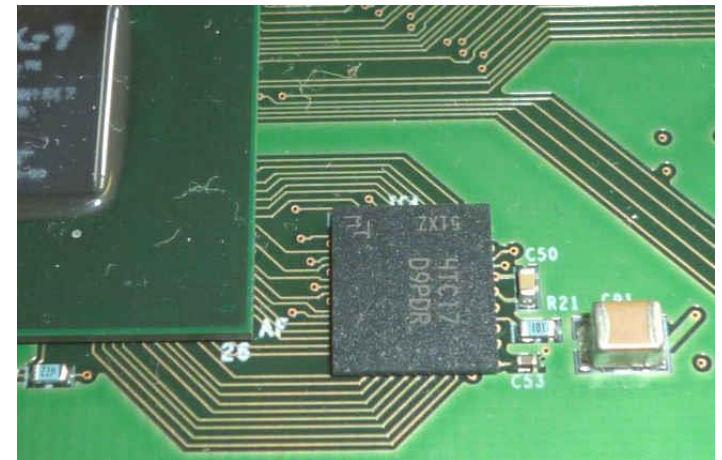
Artix7での、SDRAM用プログラム

```
constant IDELAY_VAL : integer:=12; --8; --16.
```

```
GEN1:      for i in 0 to 15 generate
IDELAY: IDELAYE2 generic map(DELAY_SRC=>"IDATAIN", IDELAY_TYPE=>"FIXED", IDELAY_VALUE=>IDELAY_VAL)
port map(IDATAIN=>mdin(i), DATAOUT=>mdin_b(i), C=>'0', CE=>'0', CINVCTRL=>'0', CNTVALUEIN=>"0000", DATAIN=>'0', INC=>'0', LD=>'0',
LDPIPEEN=>'0', REGRST=>'0');
uin: IDDR generic map (DDR_CLK_EDGE=>"SAME_EDGE_PIPELINED") port map(c=>not clk, d=>mdin_b(i), q1=>fдин(i), q2=>fдин(i+16), CE=>'1',R=>'0',S=>'0');
uout: ODDR generic map(DDR_CLK_EDGE=>"SAME_EDGE") port map(c=>not clk, d1=>fдout(i+16), d2=>fдout(i), q=>mdout(i), CE=>'1', R=>'0', S=>'0');
end generate;
```

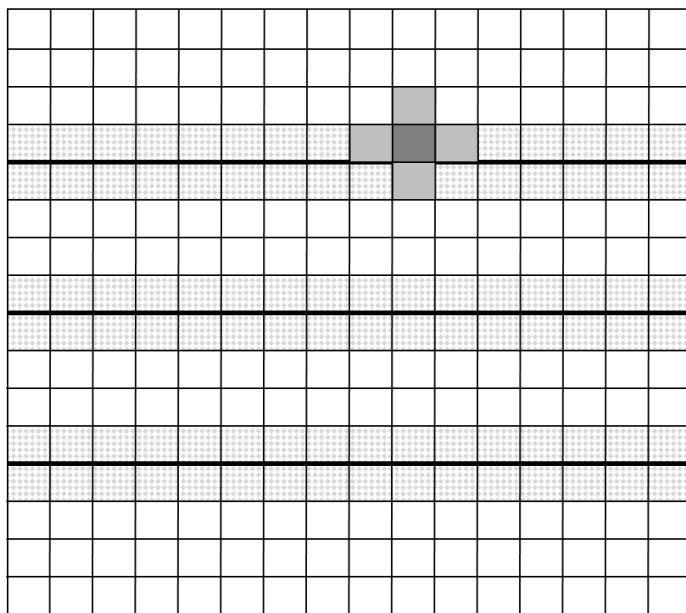


SDRAMは双方向制御になる。また、レイテンシーは使用条件で変わるので、タイミング調整が必要となる。**166Mbps**で使用。現在はDDR-メモリを使用しているが、**DDR2かそれ以上に最新化を図らなければならない。**



モジュール間送受信で重心計算

16×16のデータ配列



1モジュール

↓↑ 160bitずつ送受信

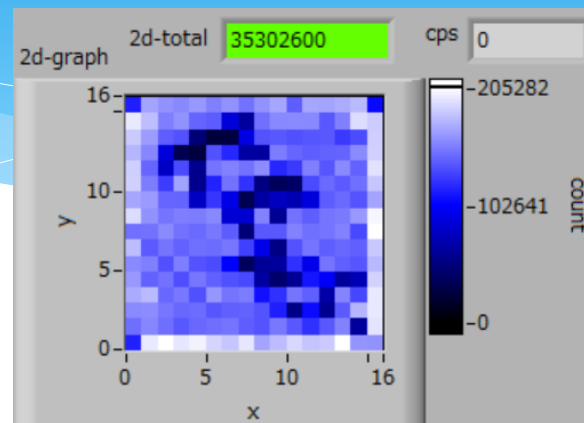
1モジュール

↓↑ 160bitずつ送受信

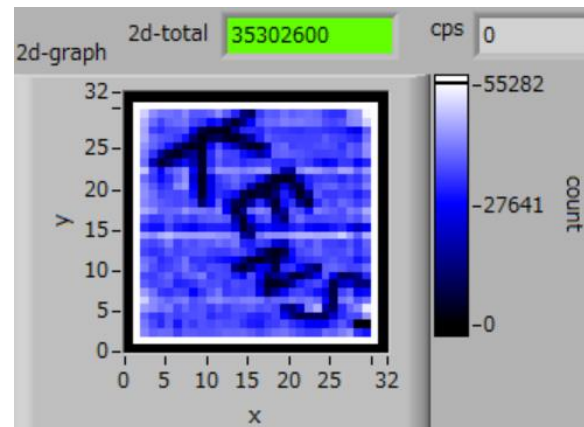
1モジュール

↓↑ 160bitずつ送受信

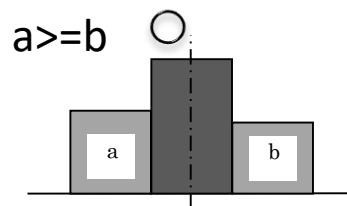
1モジュール



実配置 : 3mm分解能

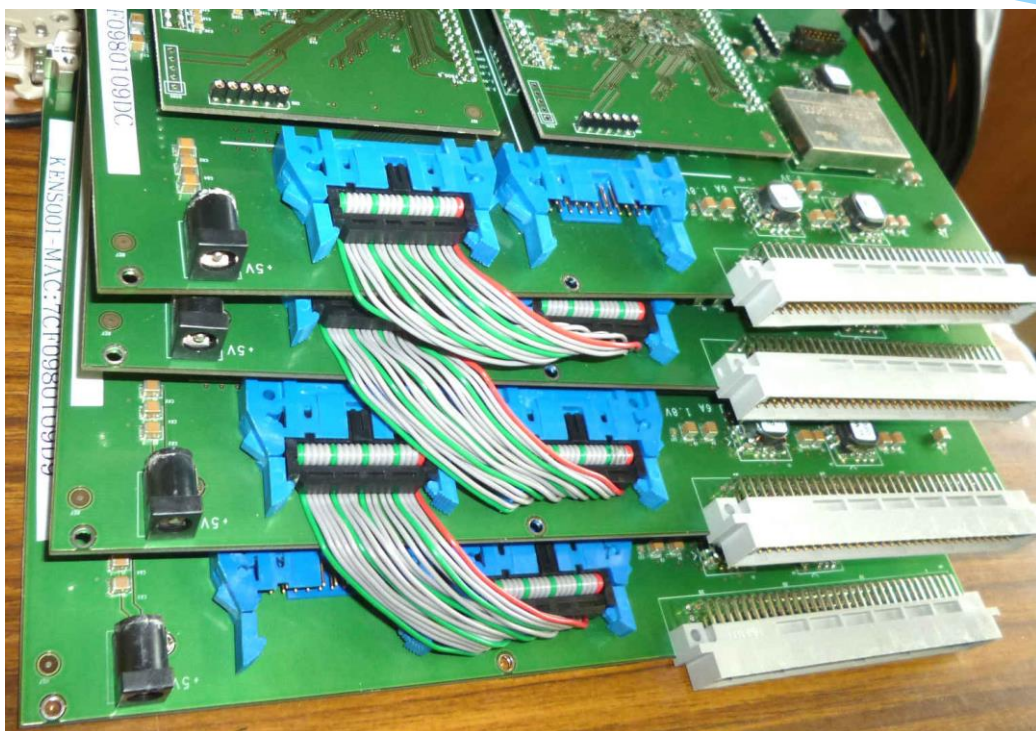


1/2重心計算 : 1.5mm分解能



重心計算のために、一番大きいピクセルと接する2ピクセルの情報を集める。

モジュール間送受信のDDR



データ評価を320nsごとに行っているため、その間に160bitのデータを送受信する必要があります。現在、1方向送受信に7芯を使用しているため、24bit/320nsとなります。信号同期に100ns程度をみると、24bit/220nsとなり、1bit/9.2nsで11Mbpsの転送が必要である。

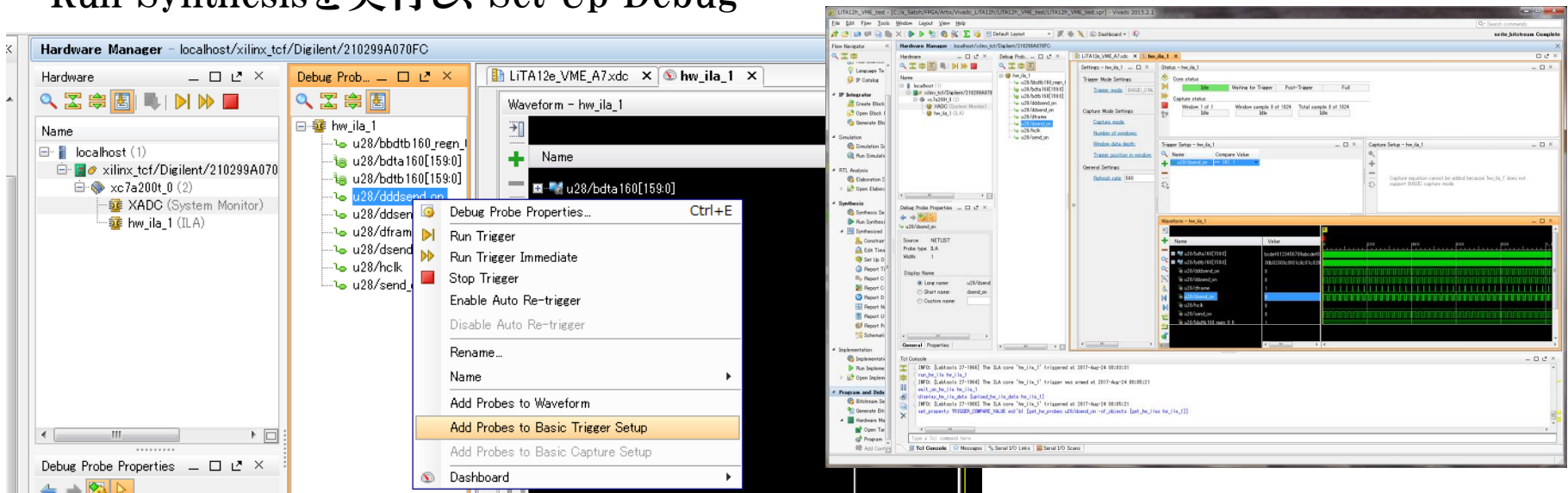
DDR解析に便利なデバッグ機能

VHDL内で見たい信号を定義

```
00
69 attribute mark_debug : string;
70 attribute keep : string;
71 attribute mark_debug of pflen : signal is "TRUE";
72 attribute mark_debug of pfdo : signal is "TRUE";
73 attribute mark_debug of ppfwen : signal is "TRUE";
74 attribute mark_debug of fdout : signal is "TRUE";
75
```

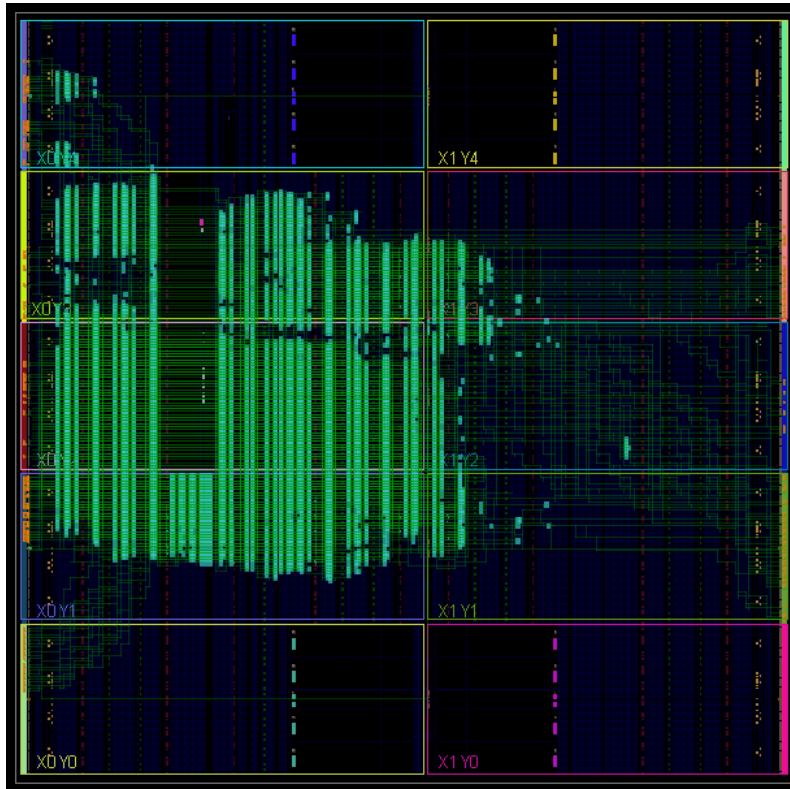
Run Synthesisを実行し、Set Up Debug

Generate Bitstreamを実行し、
FPGAに書き込む

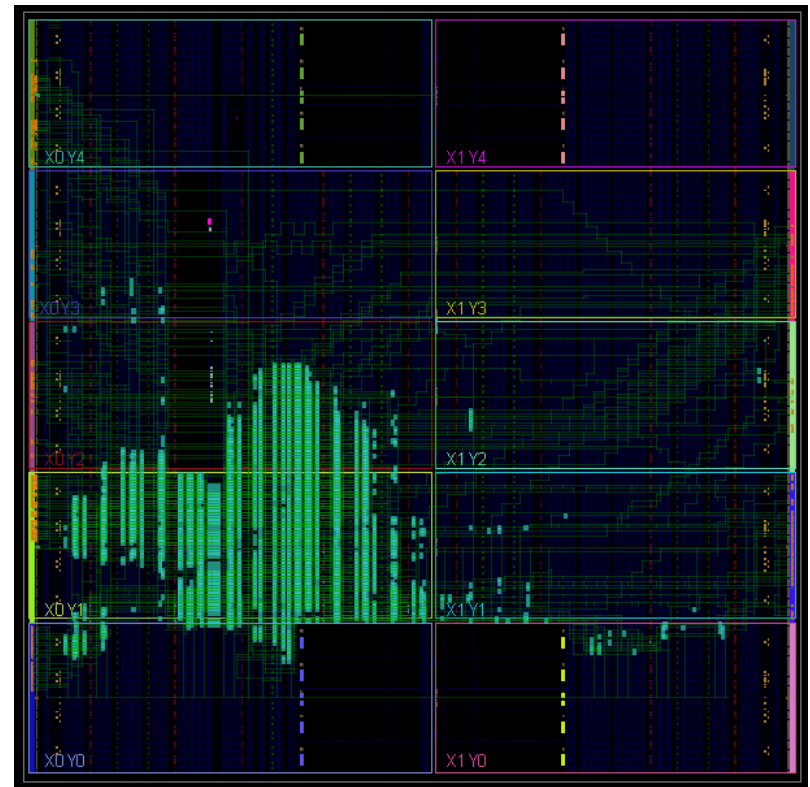


DDR解析に便利なデバッグ機能

デバッグは便利だが、
多くのビットを見ると容量が大きく、遅くなる。



デバッグ分多くのセルを使う(324bit) >>遅い、30分～



デバッグをやめると小さくなる >>速い～10分

まとめ

- ◆ 中性子検出器読み出し回路にDDR機能が欠かせない。
- ◆ ADC読出し:**500Mbps**、差動LVDS受信。XC7A100T-FGG484でADC-10bit-50MHzを32ch処理。
- ◆ SDRAM:**166Mbps**、送受信。
- ◆ モジュール間のデータ送受信:**111Mbps**、DDR送信と受信。
>>少し無謀。
- ◆ Vivadoのデバッグ機能で確認しながら試行錯誤している。