# ファクシミリの現状と将来
## -日本の伝統芸となった通信技術-
## （HardはSoftで実現可能）

画像電子学会研究会　招待講演資料
Egretcom（株）　水谷　幹男

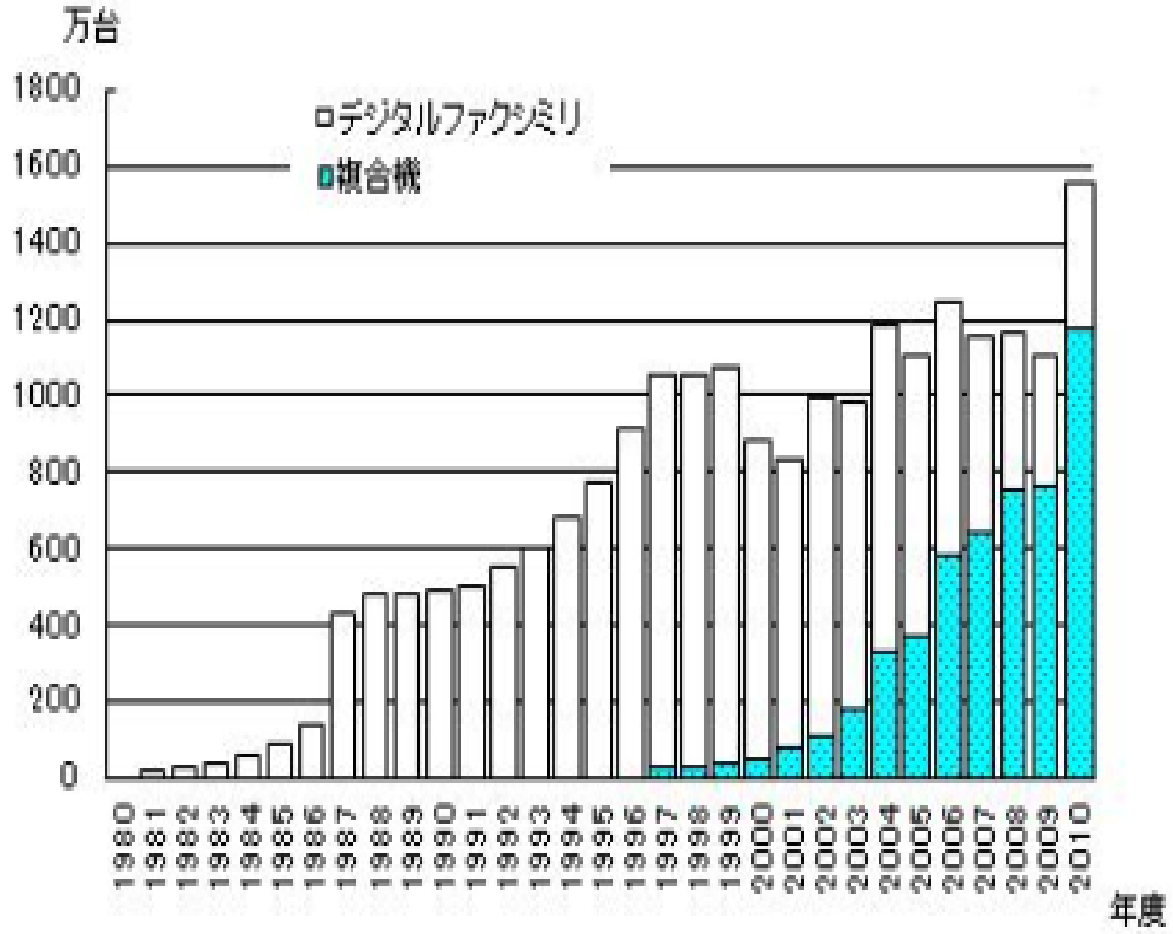
2016年3月3日

# 目次

1. ファクシミリの出荷台数と用途の変化
2. ファクシミリの構成要素
3. QAM変調の仕組み
4. 電話網のVoIP回線移行と課題
5. MATLABコーディング
6. CPU速度の進化
7. ハードモデムからソフトモデムへ

# 1. ファクシミリの出荷台数と用途の変化

# ファクシミリの出荷統計（１９８０～２０１０）



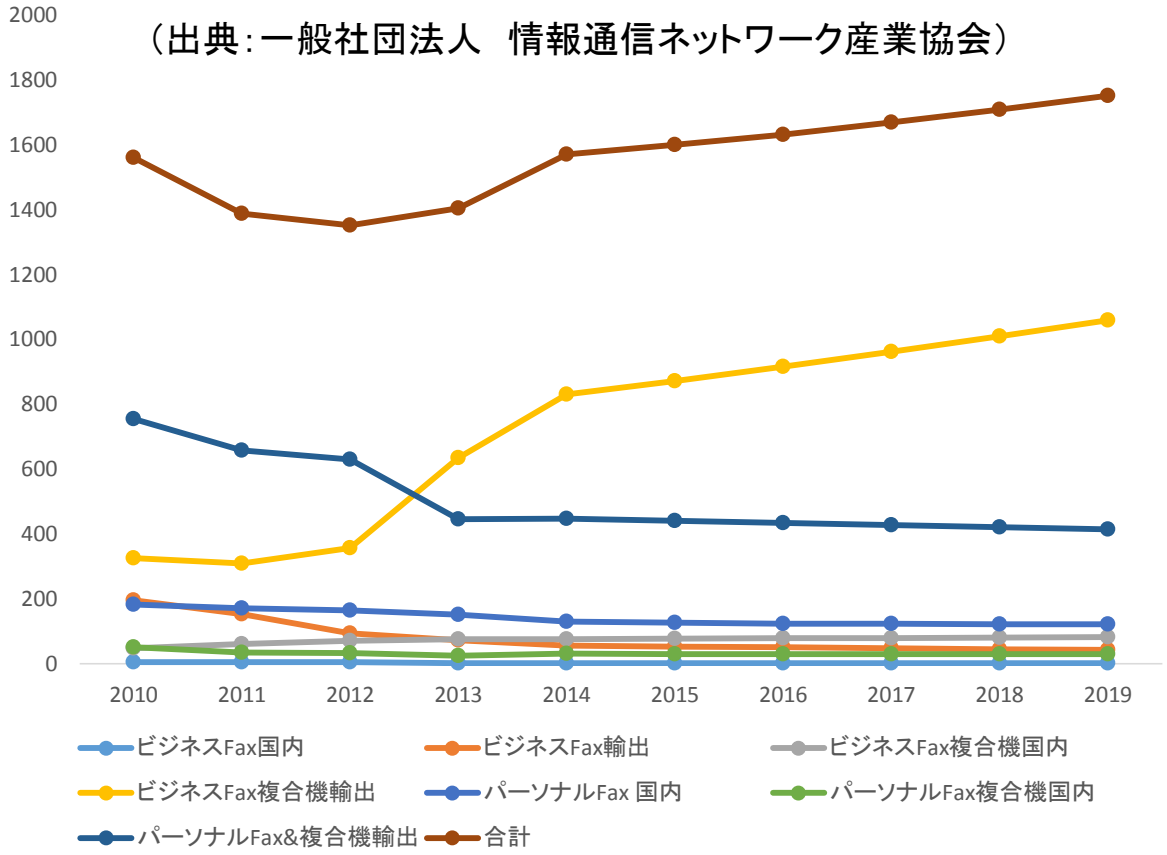

「出典：一般社団法人情報通信ネットワーク産業協会発行
通信機器中期需要予測の年度出荷実績」

**「デジタルファクシミリと複合機の出荷統計台数（含む輸出）」**

**＊）M. O. T. T 代表取締役
小川睦夫氏からの提供資料**

# 工業会統計に見るファクシミリの現状（２０１０～２０１９）

## 販売台数推移(万台)

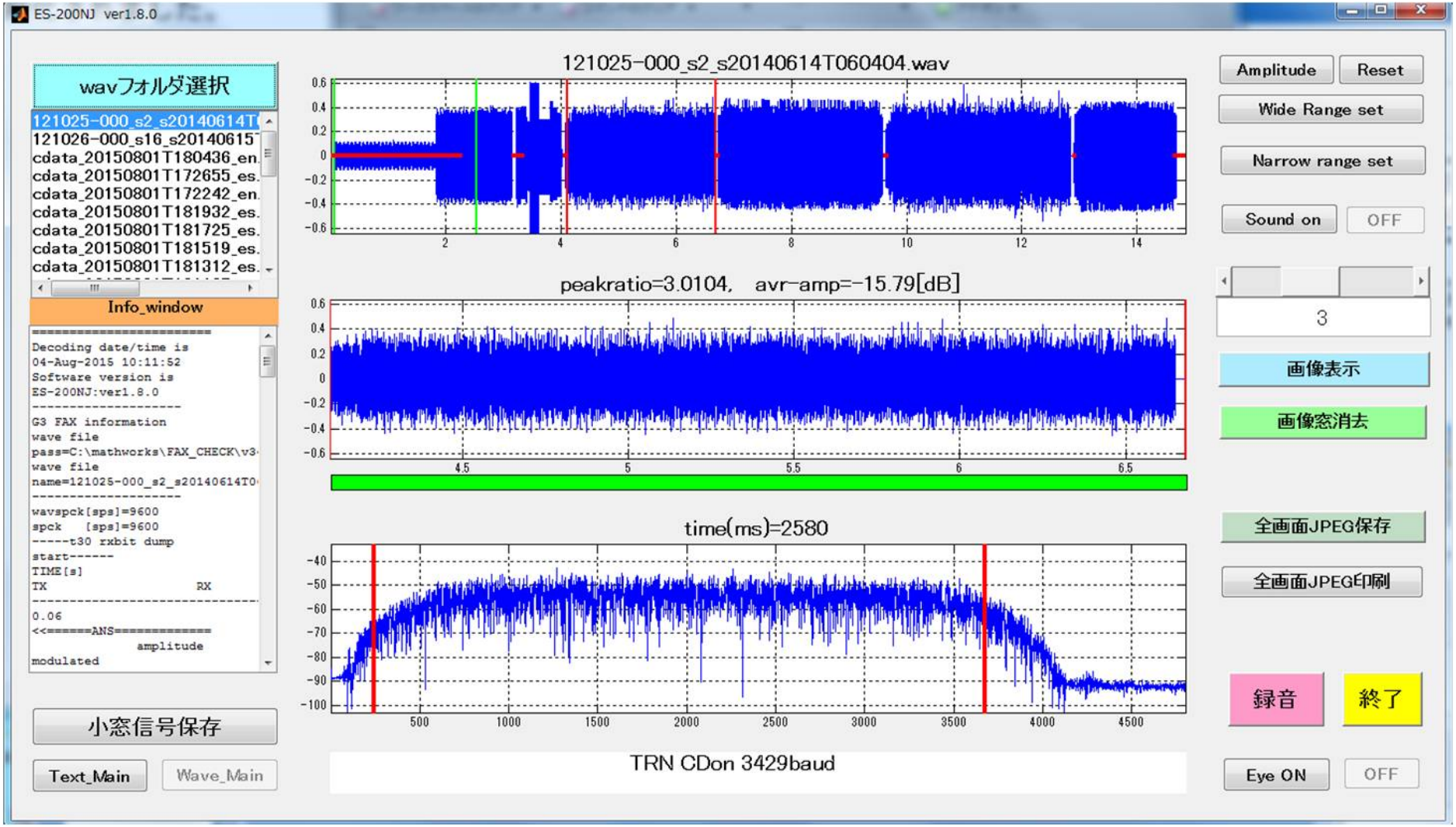



＊）M．O．T．T　代表取締役
小川睦夫氏からの提供資料

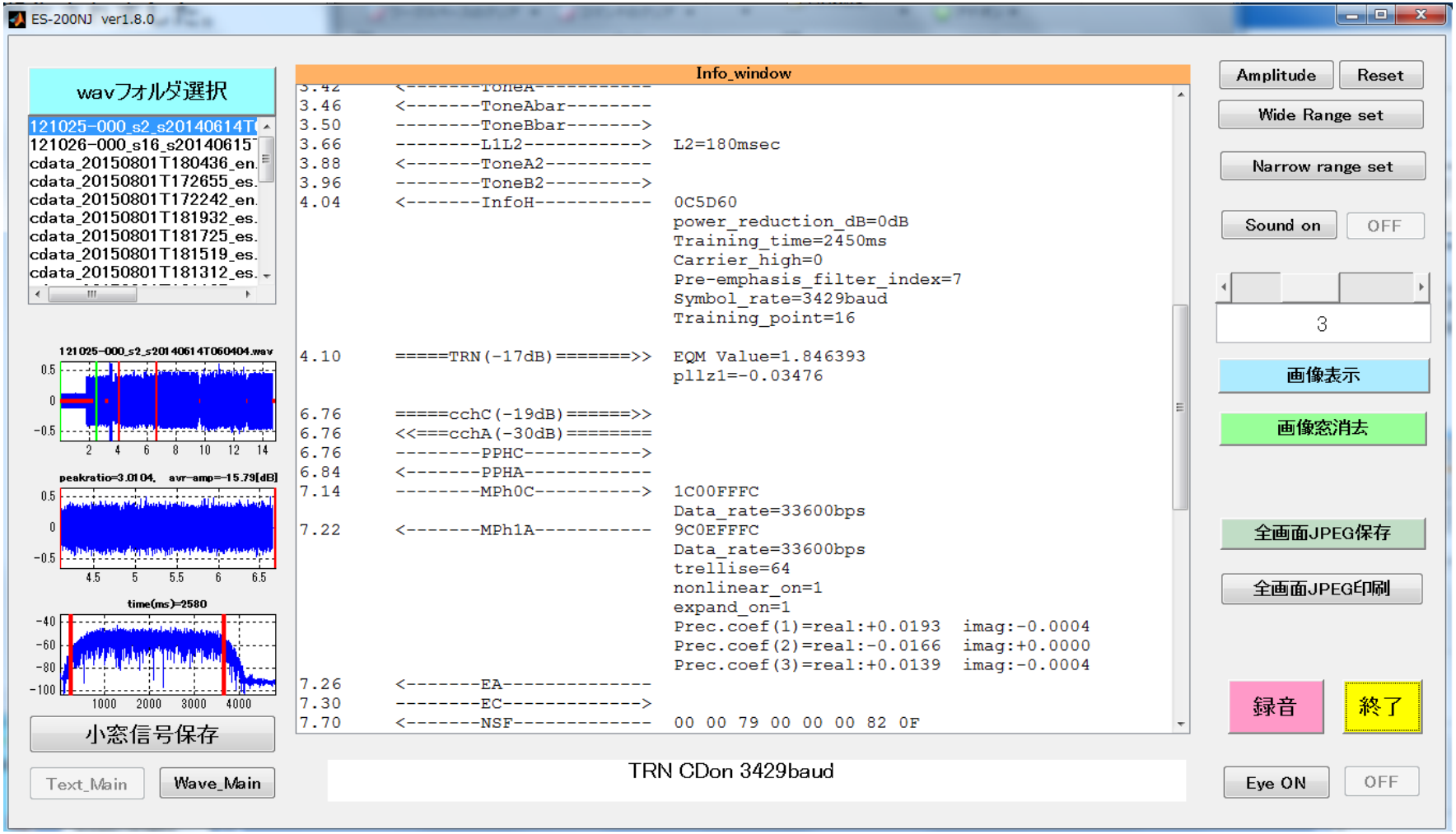
# ファクシミリの用途の変化

1. インターネットメールの普及
    - 個人用途のパーソナルFAXは減少しているが
    - 法人用途のビジネスFAX・MFPは増加の一途で、輸出も含め
      現在でも年間1600万台が出荷されている。
2. 個人情報管理の徹底
    - 顧客情報の流出を警戒
        --＞クレジットカード／電子マネーのカード発行登録
        --＞免許証のコピーをFAXで送信
3. インターネットのセキュリティ不信
    - 弁理士／弁護士の対顧客情報通信
    - 銀行／保険業務全般のFAX使用増加
    - 外交金融等官庁文書のFAX使用増加
4. ＝＝＞EgretcomへのFAX受信NGの解析依頼増加

# ＦＡＸアナライザー　ＥＳ－２００

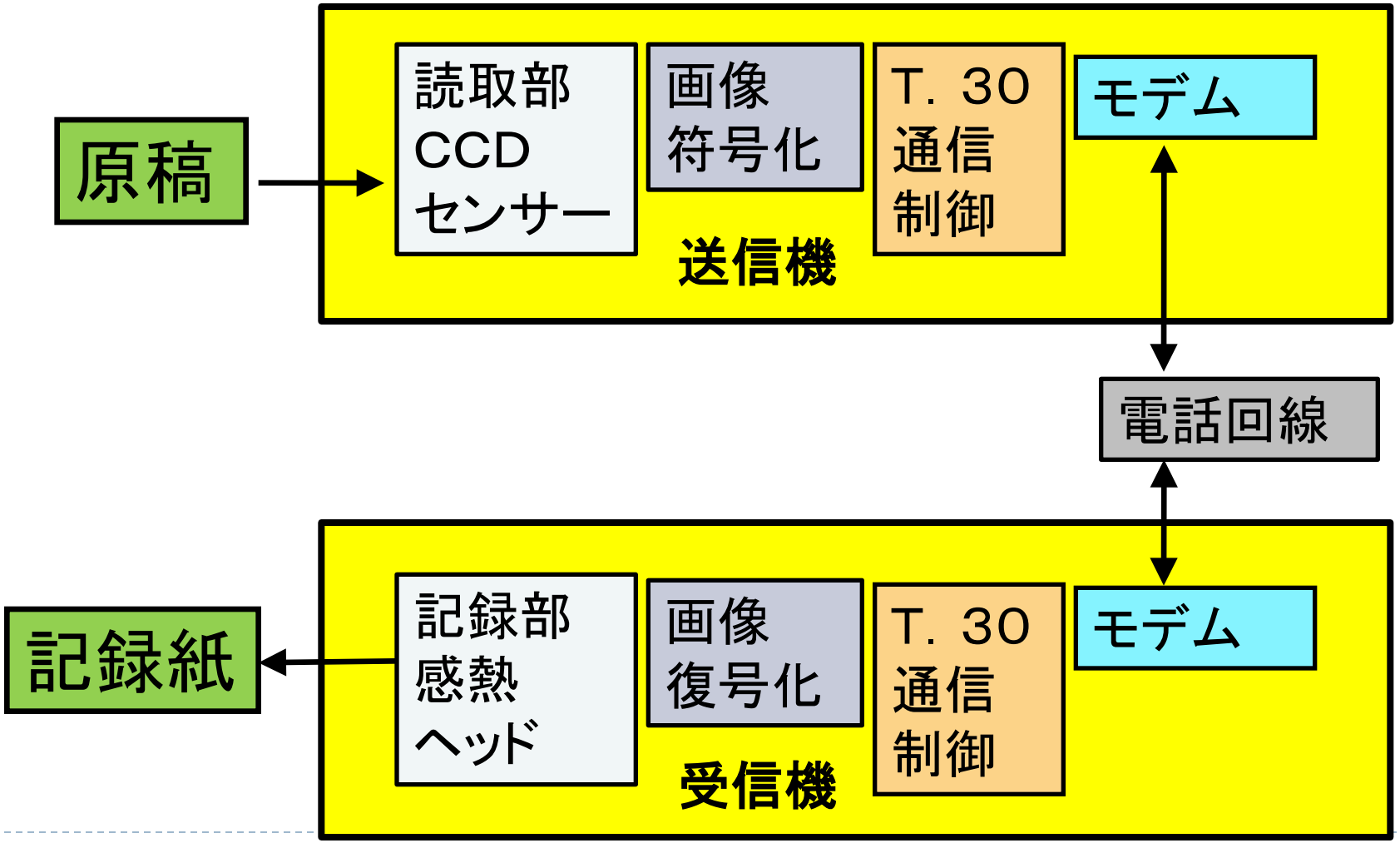


# ＦＡＸアナライザー　ＥＳ－２００



# 2. ファクシミリの構成要素

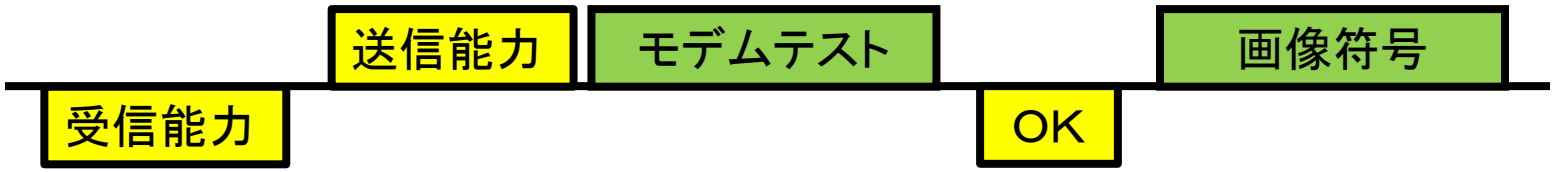
# Ｇ３－ＦＡＸの構成要素


原稿
読取部
CCD
センサー
画像
符号化
T. 30
通信
制御
モデム
送信機
電話回線
記録紙
記録部
感熱
ヘッド
画像
復号化
T. 30
通信
制御
モデム
受信機

# Ｔ．３０の革新性

＃Ｔ．30：世界初の通信プロトコル（1979年）

・通信制御情報と、画像情報をすべてディジタル化した。
・コマンド（命令）とそれへのレスポンス(応答)を定義し、確実にフローを前進させる、手順を定義した。
・当時としては、最高速の9600bpsを画像転送に使用し、命令／応答は雑音に強い300bpsを使用した。
・命令／応答は半2重手順、3way-handshake手順を基本とした。command->response->acknowledge
・データ通信規格・CCITT－MHSより6年前
・TCP／IPより7年前にプロトコルを規定していた。



# 静止画像符号化の進化

・1979年：MH　　モディファイド・ホフマン
　　1次元圧縮（NTT推奨）
　　：MR　　モディファイド・リード
　　2次元圧縮（KDD推奨）
　　4ライン終端
・1988年：MMR　モディファイド＾2・リード
　　2次元圧縮（KDD推奨）
　　無限ライン終端
・1994年：JPEG　カラー符号化
・1996年：JBIG　　白黒予測符号化

# 3. QAM変復調の仕組み

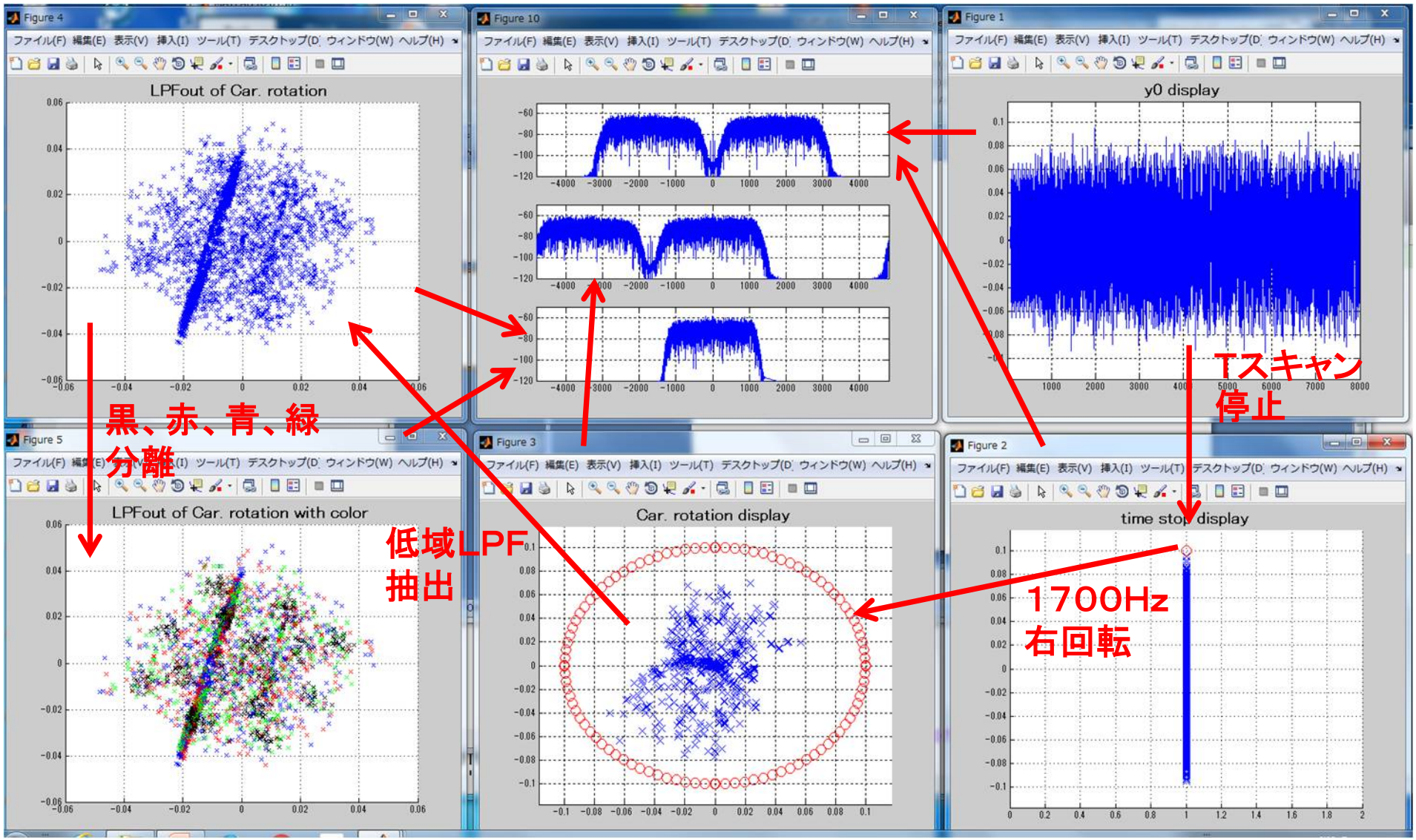


Figure 4
LPFout of Car. rotation
Figure 10
Figure 1
y0 display
黒、赤、青、緑
分離
Tスキャン
停止
Figure 5
LPFout of Car. rotation with color
低域LPF
抽出
Figure 3
Car. rotation display
Figure 2
time stop display
1700Hz
右回転

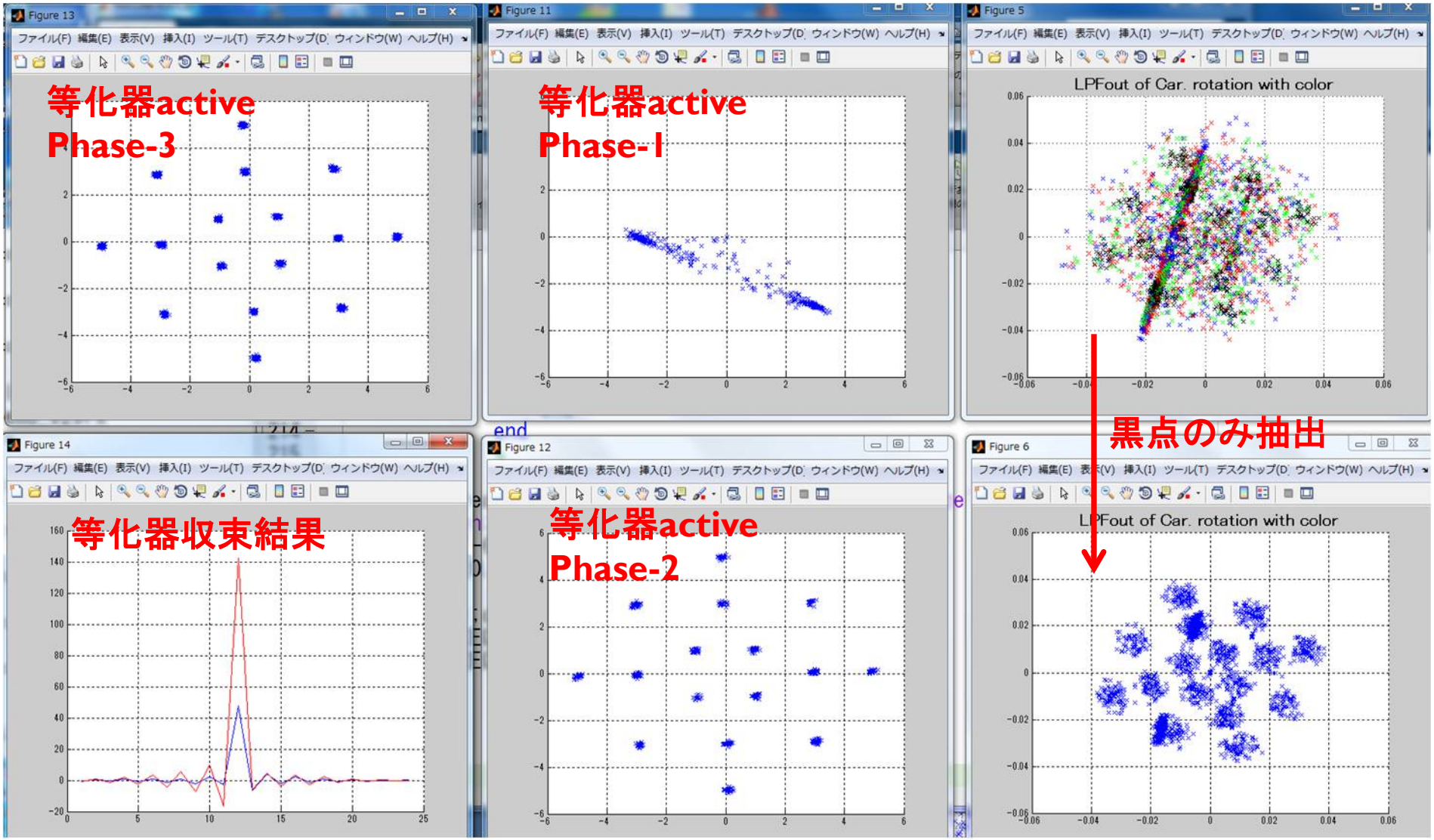
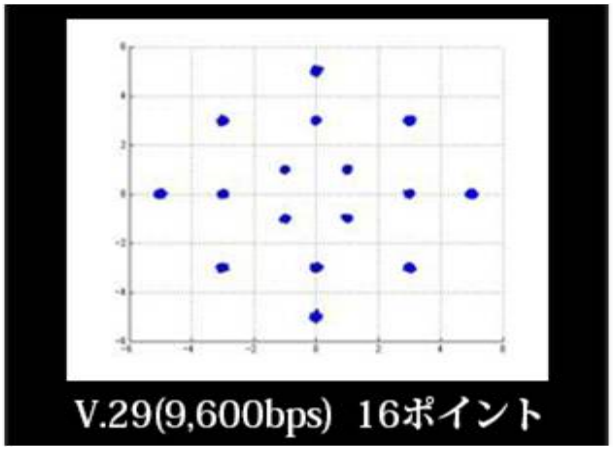
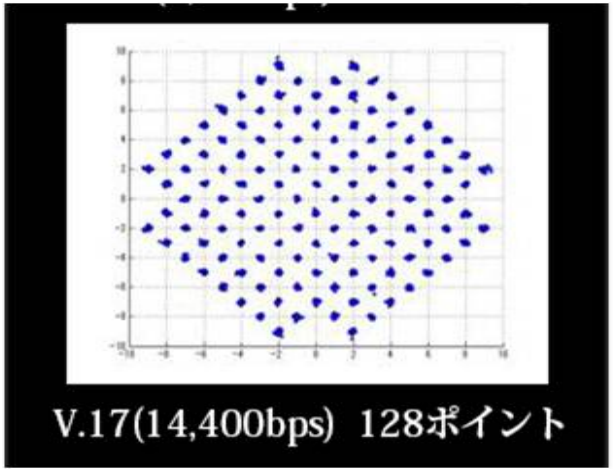


Figure 13
等化器active
Phase-3
Figure 11
等化器active
Phase-1
Figure 5
LPFout of Car. rotation with color
黒点のみ抽出
Figure 14
等化器収束結果
Figure 12
等化器active
Phase-2
Figure 6
LPFout of Car. rotation with color

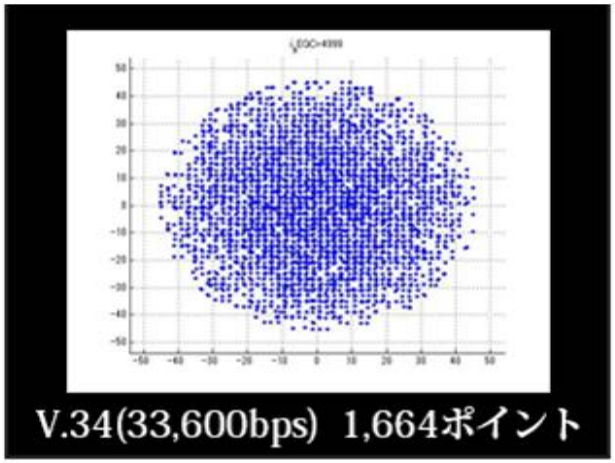
# 各種モデム星座



V.29(9,600bps) 16ポイント

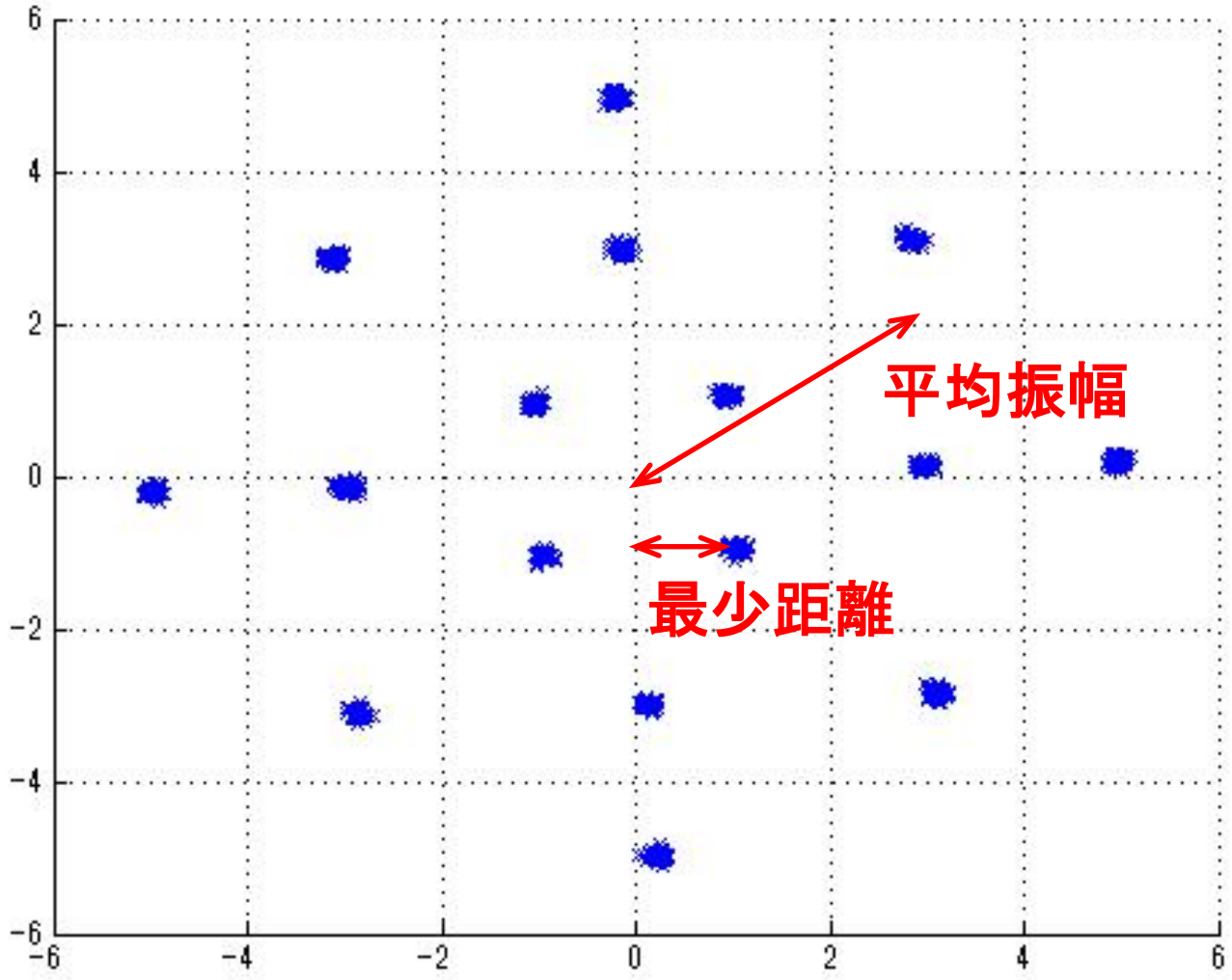


V.17(14,400bps) 128ポイント



V.34(33,600bps) 1,664ポイント

# 限界SN＝20＊LOG10(平均振幅／最少距離)



# モデム星座と限界ＳＮ
# ビット誤り率＝10^(-6)で比較

| Vxx-bps | 限界SN[dB] | Vxx-bps | 限界SN[dB] |
|---|---|---|---|
| V27-2400 | 12.042 | V34-14400 | 22.042 |
| V27-4800 | 25.474 | V34-16800 | 24.559 |
| V29-7200 | 16.434 | V34-19200 | 26.498 |
| V29-9600 | 20.334 | V34-21600 | 28.658 |
| V17-9600 | 22.042 | V34-24000 | 30.579 |
| V17-12000 | 25.263 | V34-26400 | 32.910 |
| V17-14400 | 28.203 | V34-28800 | 35.131 |
| | | V34-31200 | 36.892 |
| | | V34-33600 | 39.280 |

# 4. 電話網のVoIP回線移行と課題

# 電話網がＶｏＩＰ回線に入れ替わる

1．2020年までに、NTTの電話回線はすべて、IP電話網（VoIP回線）に入れ替わる。

2．末端のVoIPーTAが、アナログ信号を、μLawー64Kのディジタル信号に変換し、20m秒単位のRTPパケットとして、IP電話網で飛んでいく。

3．　μLawー64Kで可能なモデム最高速度は、31．2Kbpsであるため、33．6KbpsのスーパーG3FAXは、いつも速度フォールバックが発生する。

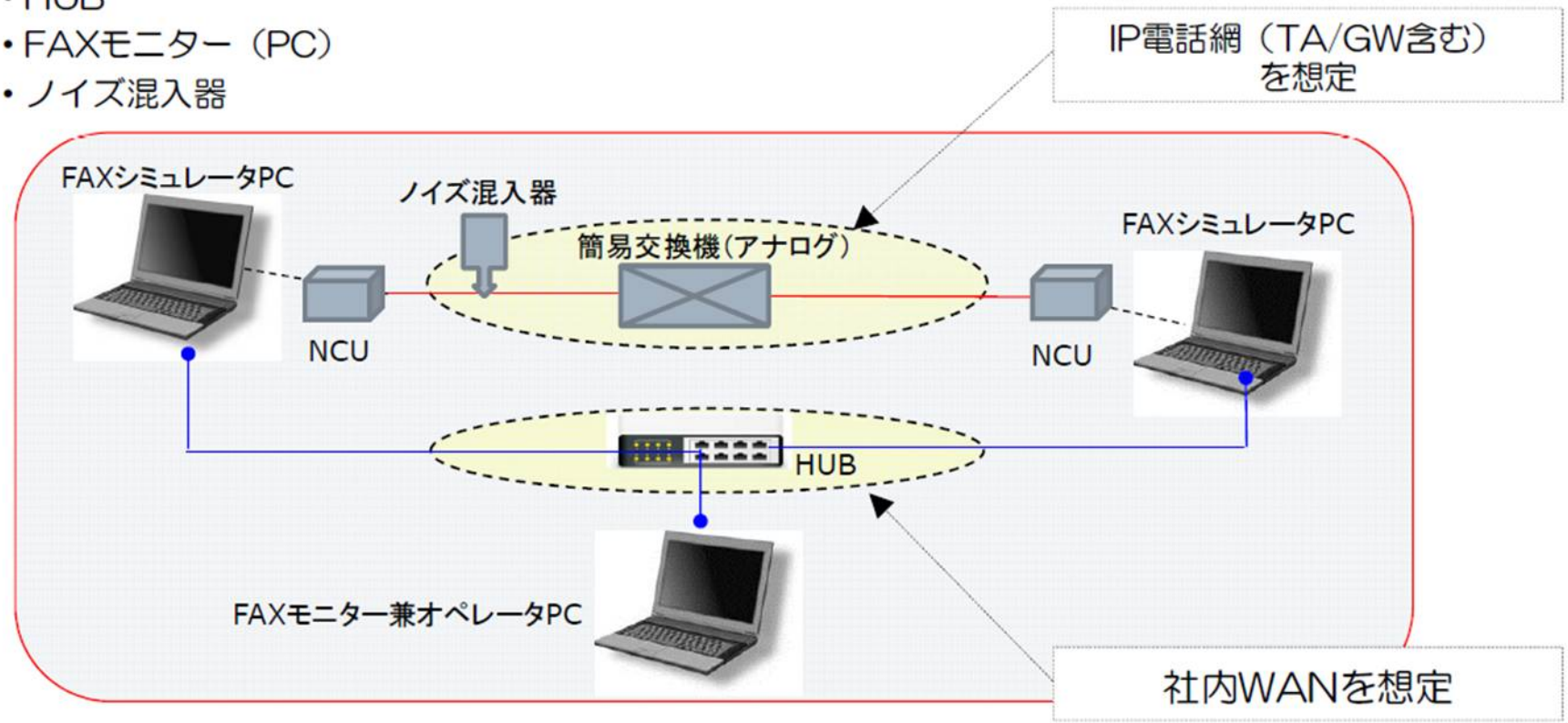
4．交換網が輻輳すると、20m秒単位でパケットロスが発生する。

5．VoIPーTAの水晶ズレで、jitter-bufferがオーバーフロウまたはアンダーフロウを起こし、これもパケットロスが発生する。

6．2016年4月から、総務省通達により、キャリアが固定電話番号振りだしが出来る為には、FAX通信テストが義務付けられた。

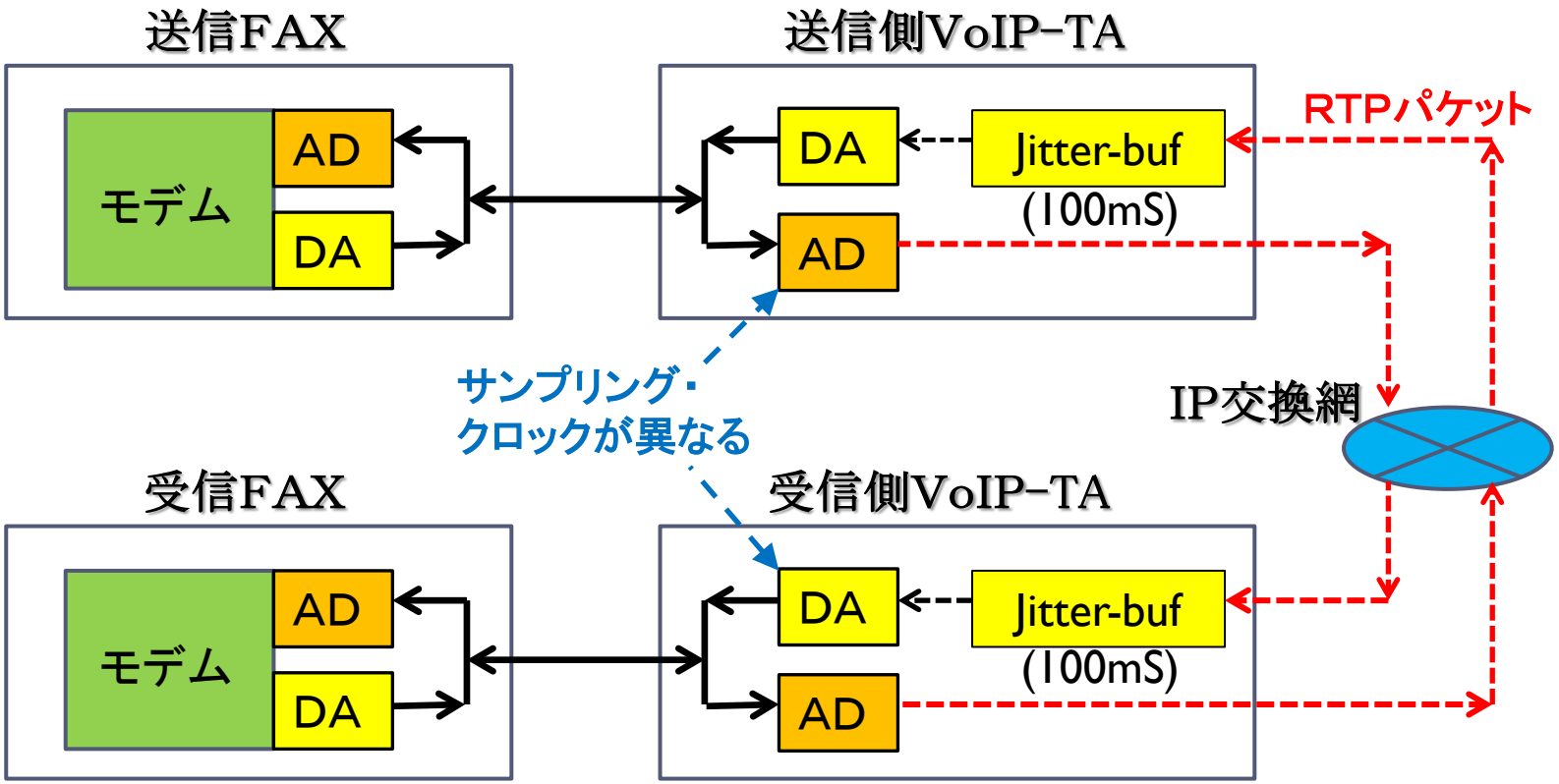
# ＶｏＩＰ回線ーＦＡＸ疎通確認システム

- FAXシミュレータ（PC）＋D/A変換機能有りNCU
- 簡易交換機（アナログ）
- HUB
- FAXモニター（PC）
- ノイズ混入器



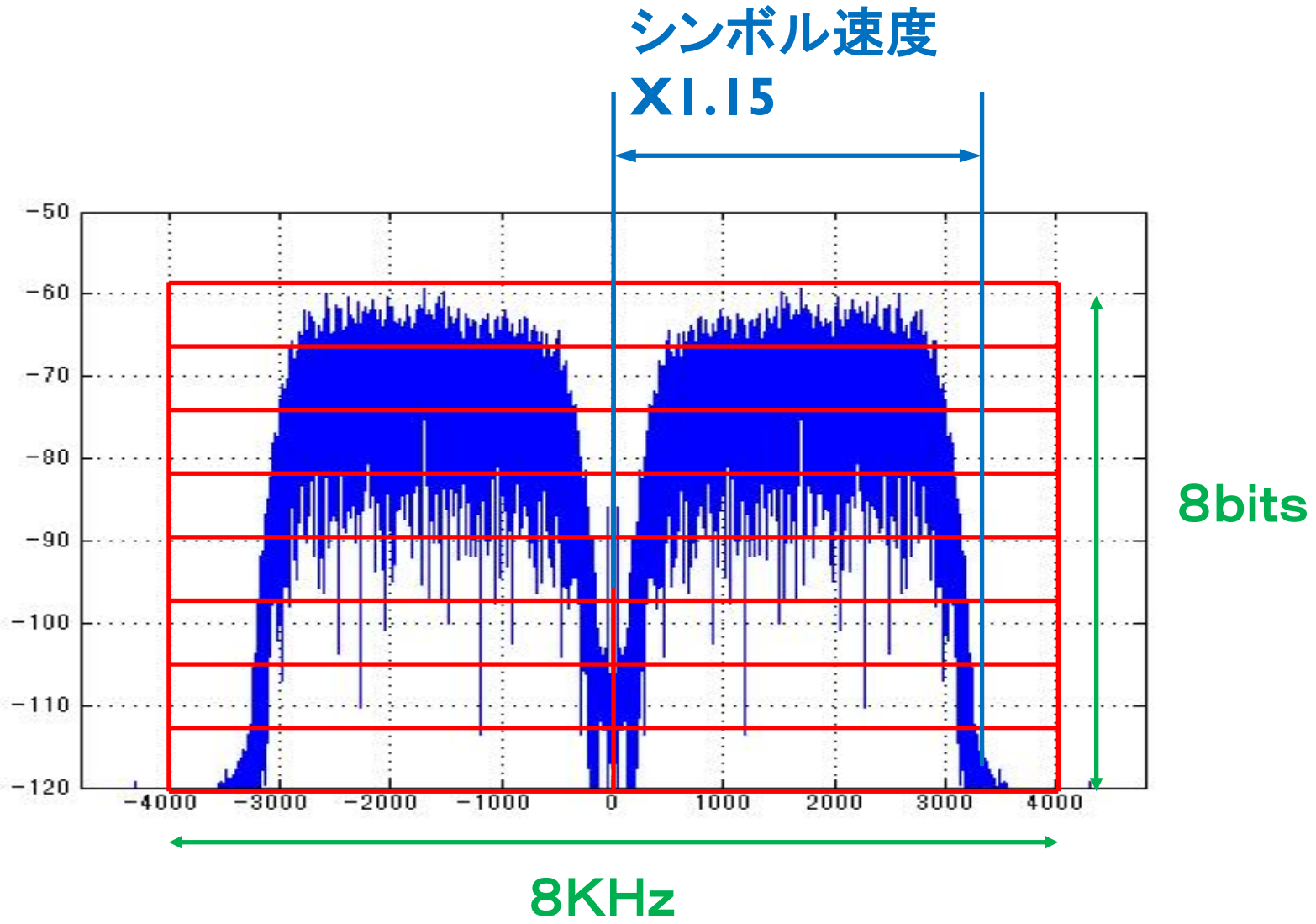
# ＶｏＩＰ回線の問題

VoIP回線でのFAX通信には、4個のDAと4個のADが関係する




送信ADと受信DAのサンプリングクロックが異なるのでJitter-bufで
オーバーフロウまたはアンダーフロウが発生し、パケット損失が発生

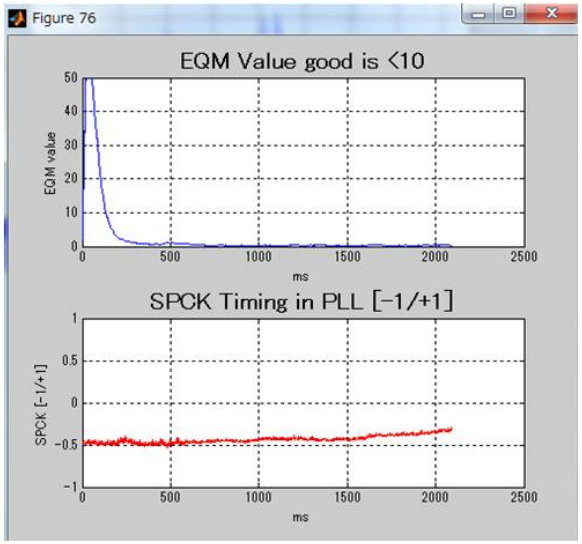
# μ－Ｌａｗ６４Ｋの限界


シンボル速度
X1.15
-50
-60
-70
-80
-90
-100
-110
-120
-4000
-3000
-2000
-1000
0
1000
2000
3000
4000
8bits
8KHz

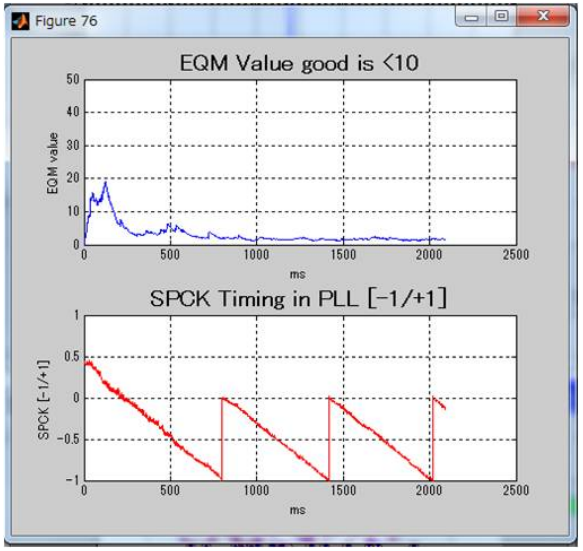
# μLaw－６４ｋ回線の通信容量（限界ｂｐｓ）
# =シンボル速度＊１．１５＊８（ｂｐｓ）

| Vxx | シンボル速度 | 限界bps |
|---|---|---|
| V27-2400 | 1200 | 11,040 |
| V27-4800 | 1600 | 14,720 |
| V29 | 2400 | 22,080 |
| V17 | 2400 | 22,080 |
| V34 | 3429 | 31,547 |
| | 3200 | 29,440 |
| | 3000 | 27,600 |
| | 2800 | 25,760 |
| | 2743 | 25,236 |
| | 2400 | 22,080 |

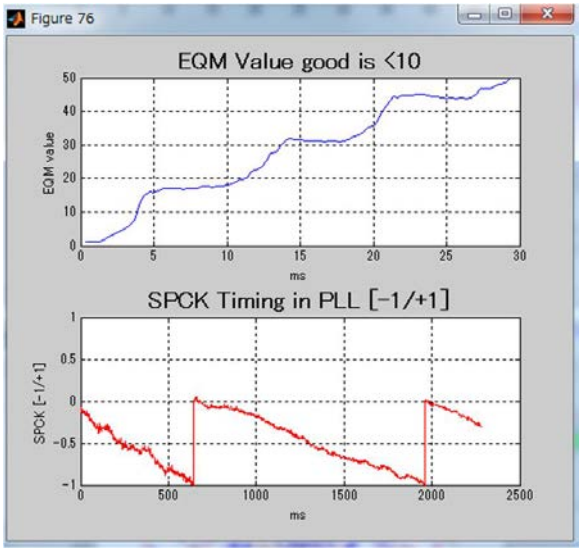
# 水晶ズレとPhase3受信中のＥＱＭ変動




水晶ズレ
PLL安定
0ppm



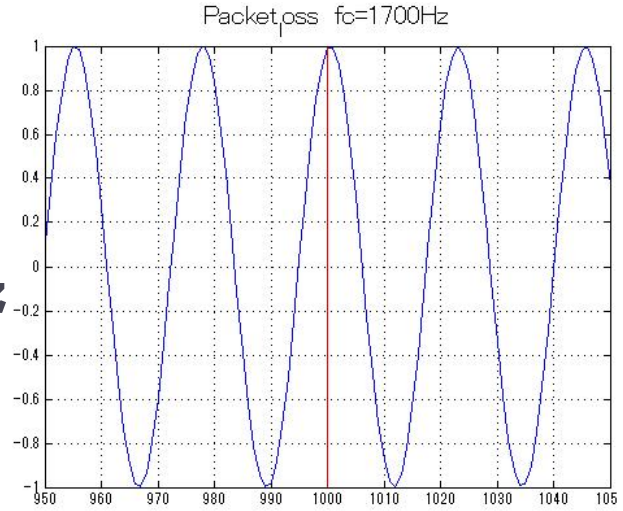

水晶ズレ
PLL安定
190ppm

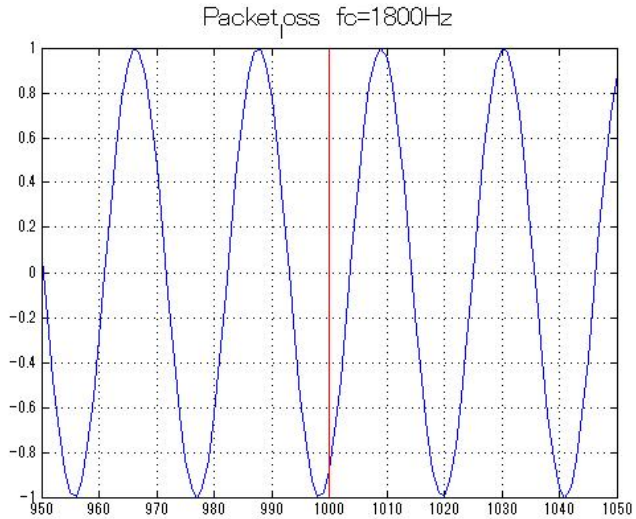



水晶ズレ+
300Hz雑音
PLL不安定
80ppm？

# 20msパケットロスによる、キャリア変化
# ＝50Hzの整数倍キャリアは、位相連続

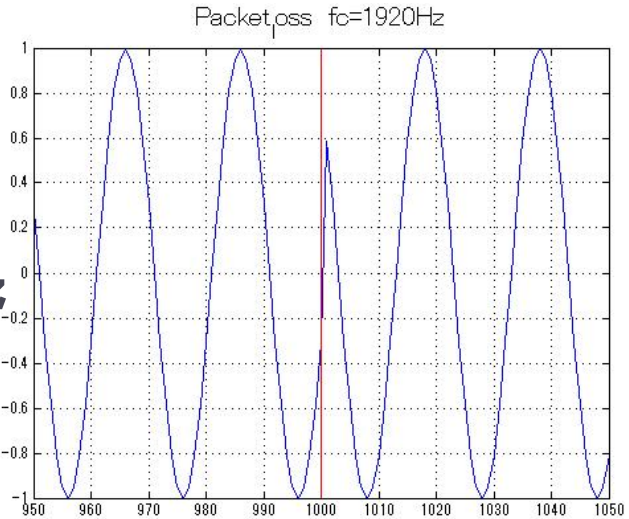
V２９
1700Hz
連続

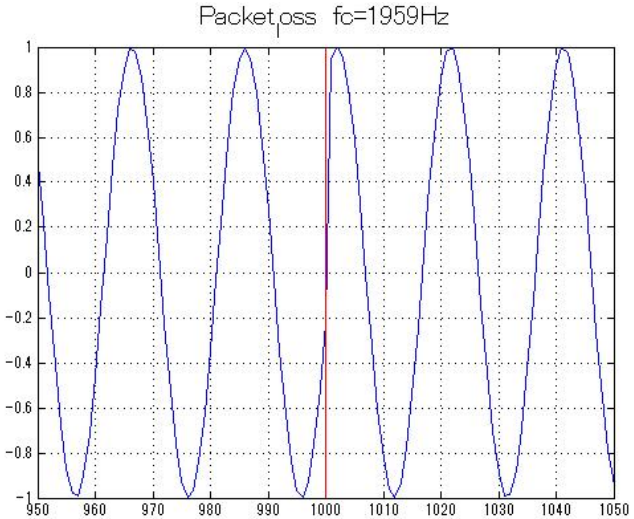



V１７
1800Hz
連続

V３４
1920Hz
不連続



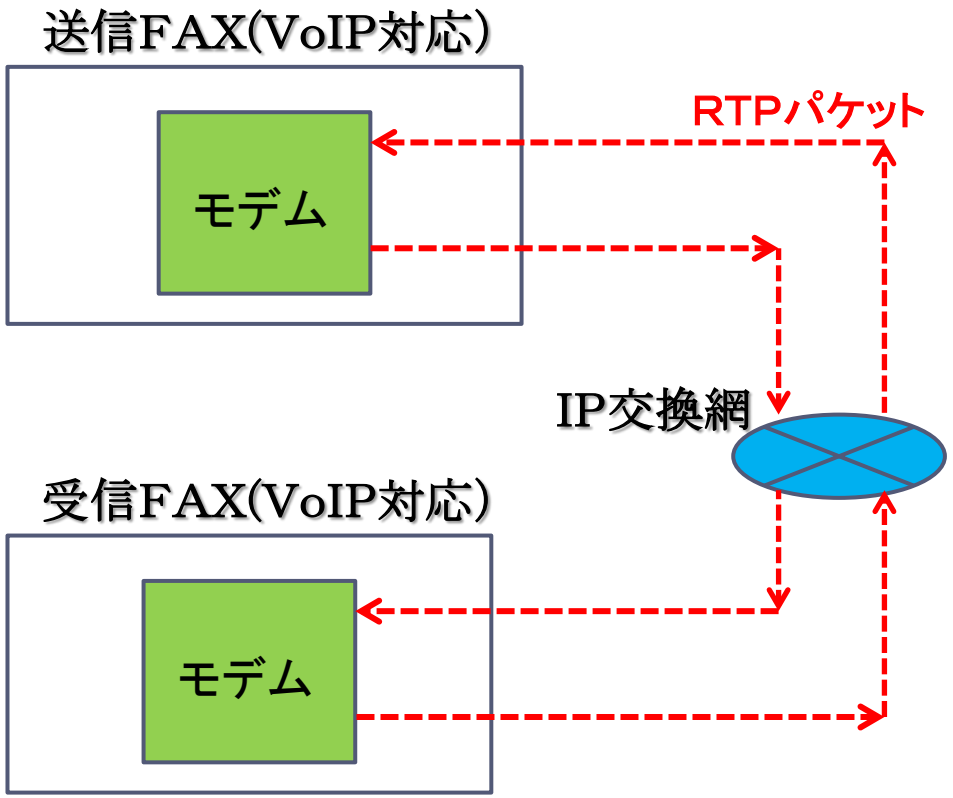

V３４
1959Hz
不連続

# ２０ｍｓパケットロスに強いモード
# =赤：Symbol/Carr.共に５０Ｈｚの整数倍

| ↓Symbol rate | Low-Carr. Low-end | Low-Carr. Center | Low-Carr. High-end | High-Carr. Low-end | High-Carr. Center | High-Carr. High-end |
|---|---|---|---|---|---|---|
| V27-1200 | 1200 | 1800 | 2400 | | | |
| V27-1600 | 1000 | 1800 | 2600 | | | |
| V29-2400 | 500 | 1700 | 2900 | | | |
| V17-2400 | 600 | 1800 | 3000 | | | |
| V34-2400 | 400 | 1600 | 2800 | 600 | 1800 | 3000 |
| V34-2743 | 274 | 1646 | 3017 | 457 | 1829 | 3200 |
| V34-2800 | 280 | 1680 | 3080 | 467 | 1867 | 3267 |
| V34-3000 | 300 | 1800 | 3300 | 500 | 2000 | 3500 |
| V34-3200 | 229 | 1829 | 3429 | 320 | 1920 | 3520 |
| V34-3429 | 224 | 1959 | 3673 | | | |

# ＶｏＩＰ回線の問題の解消方法

VoIPーTAをスルーして、4個のDAと4個のADを全て撤廃する。



# 5. MATLABコーディング

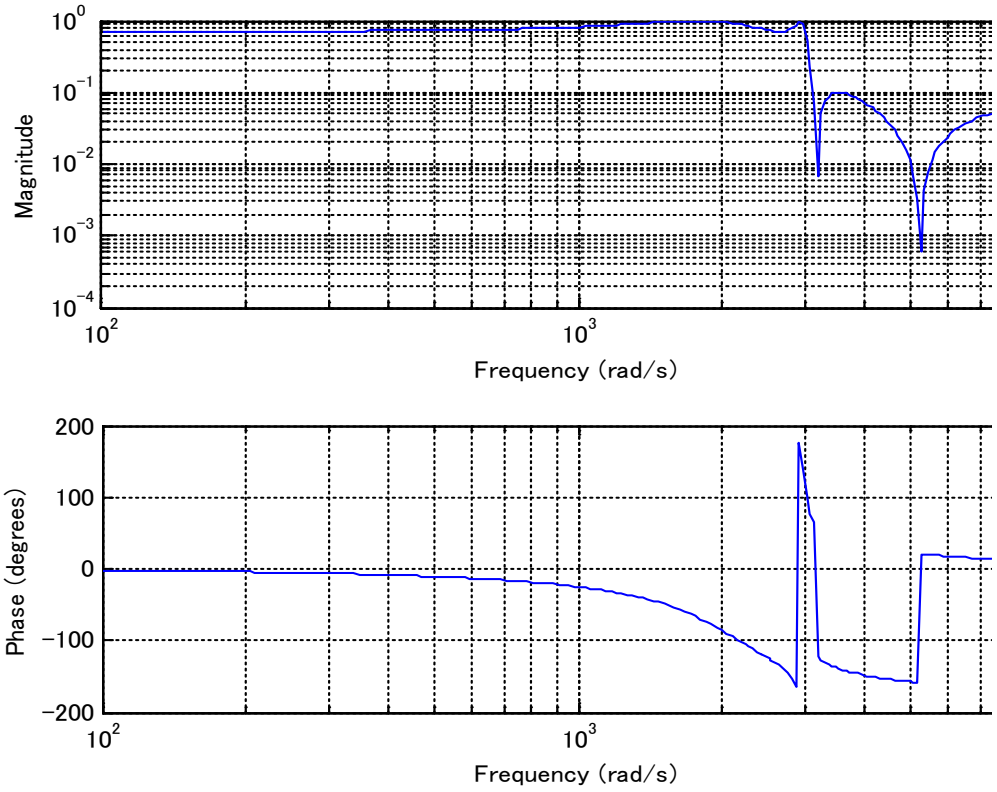
# ＭＡＴＬＡＢ活用のメリット

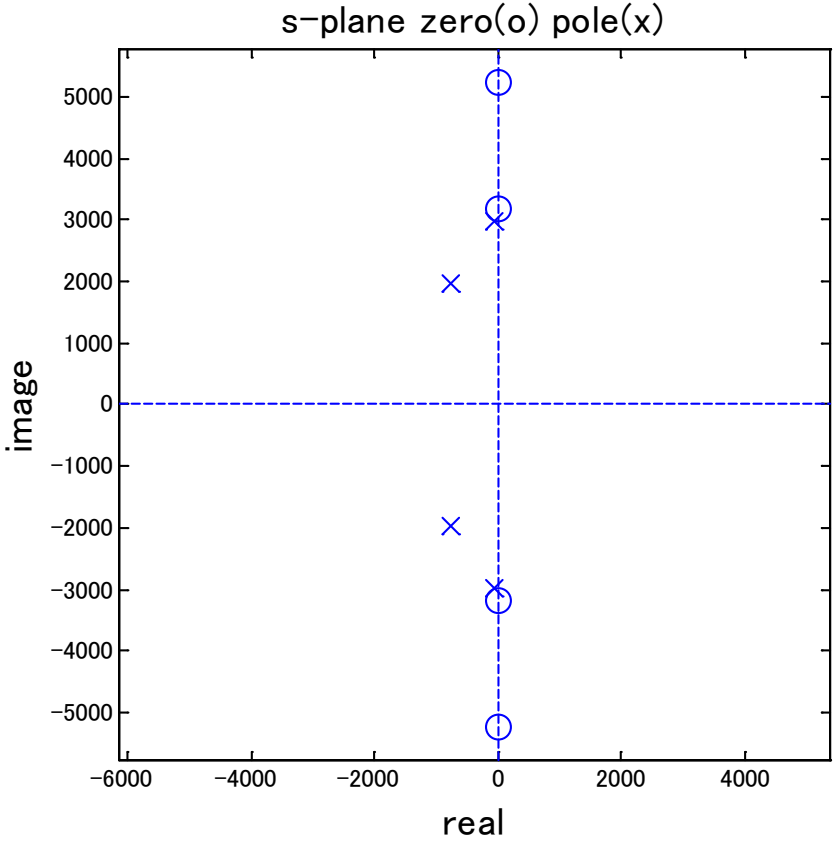
- MATLAB言語の特徴
  ＃型宣言／配列宣言不要
  ＃変数のデフォルト=倍精度複素多次元配列の
  　　ダイナミックアロケーション
  ＃インタープリター動作（シミュレーション）
  　　＝＞C言語変換（組み込みソフト）
  　　モデルベース開発が容易
- ライブラリー関数の充実
  ＃Digital信号処理ライブラリー、通信基本ブロックライブラリー、
  　IPネットワークライブラリー
  ＃PLOT関数の充実ーー＞信号図形表示が1行
  ＃GUIが、VC／JAVA並みに容易

# S変換のｆｉｌｔｅｒ特性表示例

[b,a]=ellip(4,3,20,3000,'s');

freqs(b,a); splane(b,a) ;

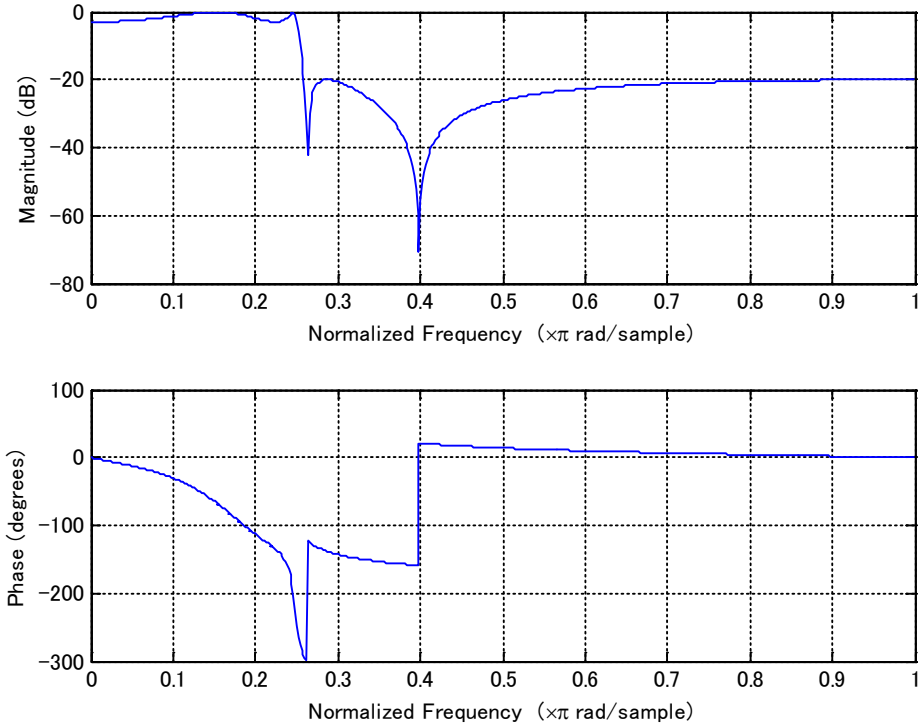


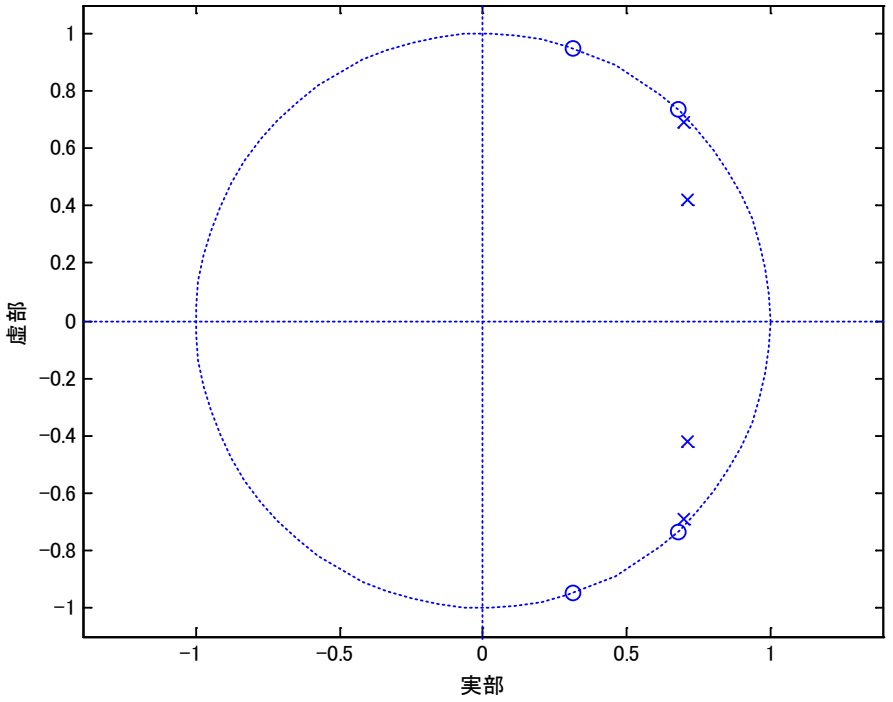



# Z変換のｆｉｌｔｅｒ特性表示例

[b,a]=ellip(4,3,20,3000/12000);

freqz(b,a); zplane(b,a) ;

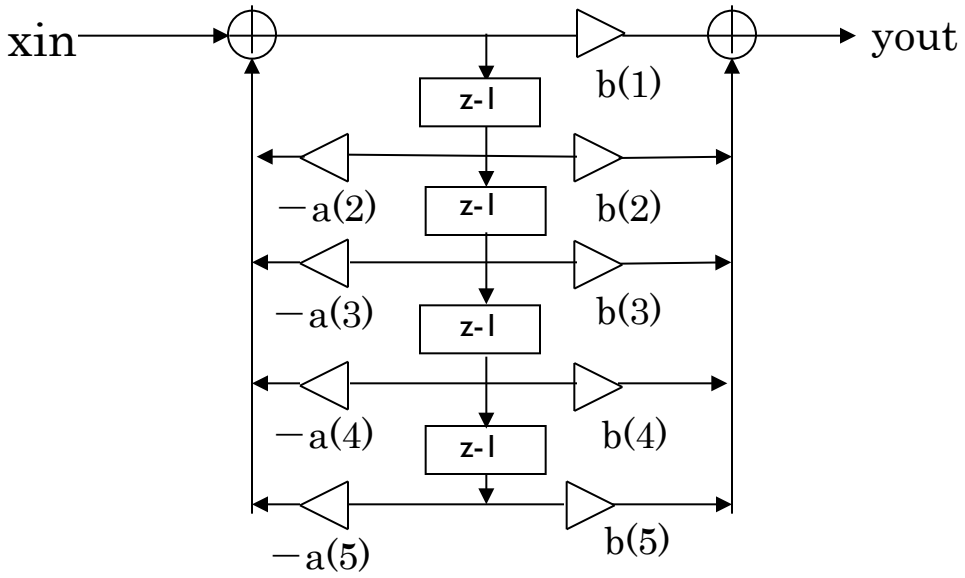


# Z変換後のZ関数の一般形
## ディジタル回路のtransfer-function

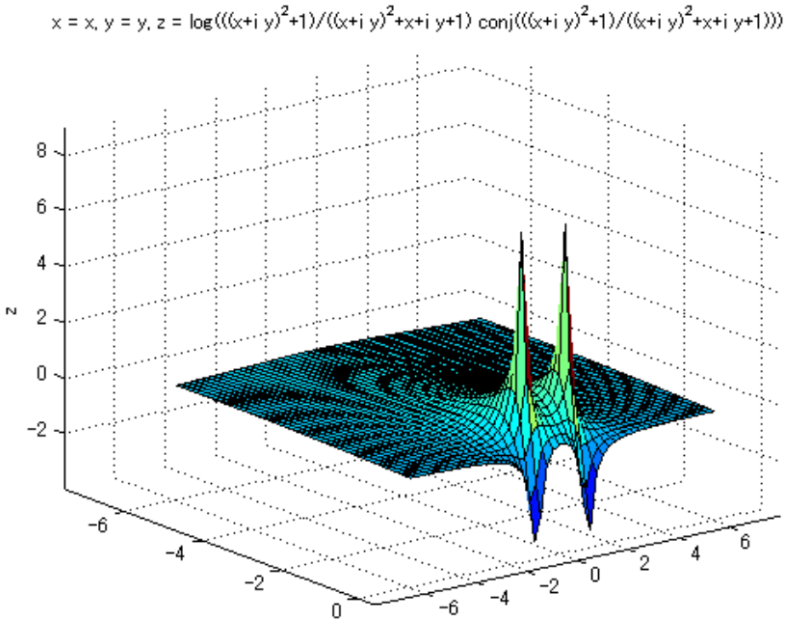
**b**＝[ b(1)　b(2)　・・・・・・・　b(n+1)]

**a**＝[　1　a(2)　・・・・・・・　a(n+1)]



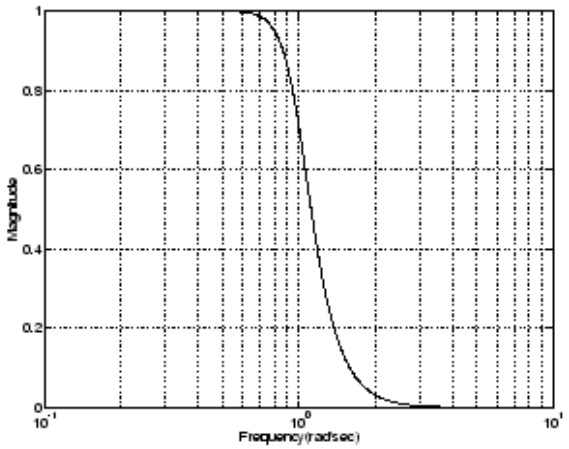
# ２次フィルター特性の３次元表示方法

```
syms s x y t
real(x);real(y);s=x+i*y;
f=(s^2+1)/(s^2+s+1);
ezsurf(x,y,10*log(f*f'),[-6,0,-6,6])
Rotate3d on
```

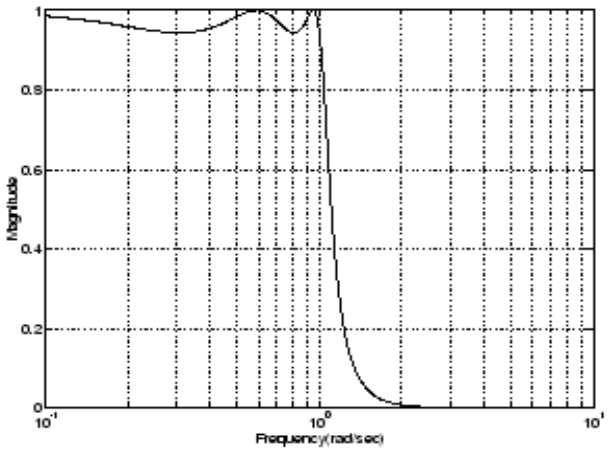


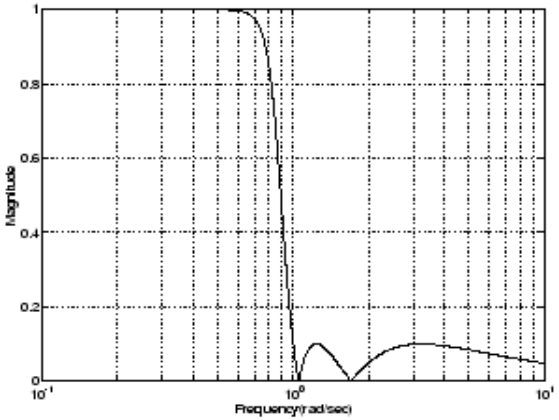
# フィルター４Ｔｙｐｅ

Butterwoth

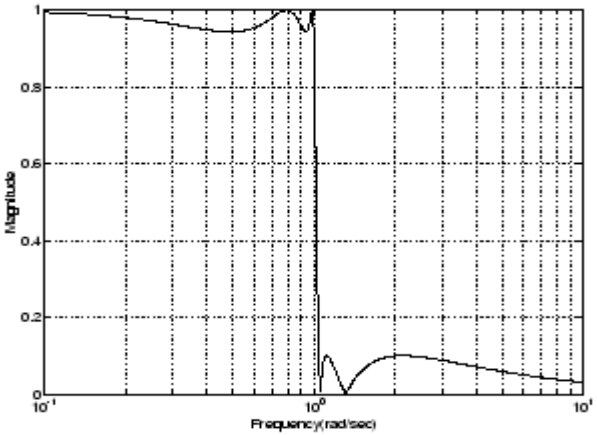



Chebyshev 1



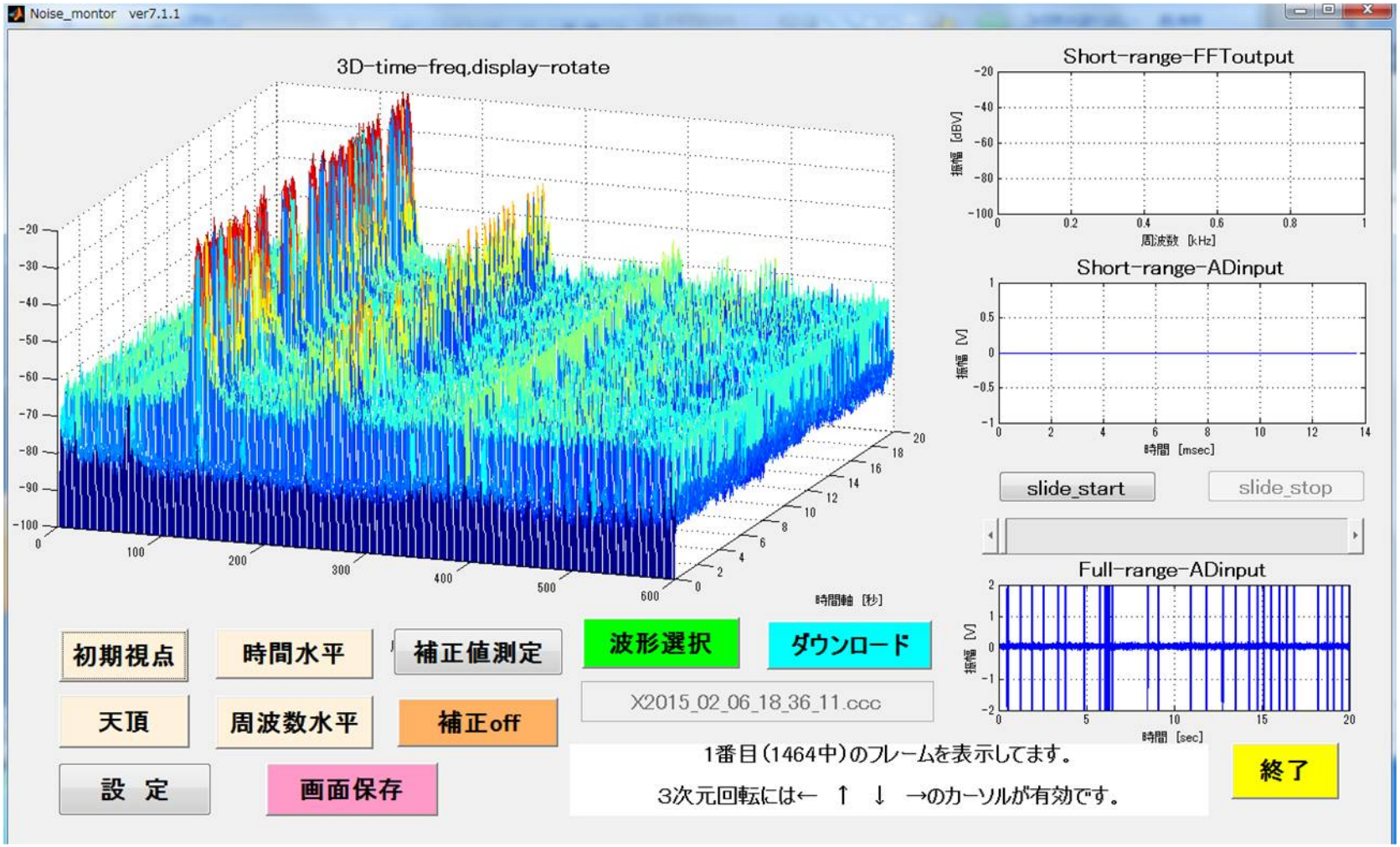




Chebyshev 2

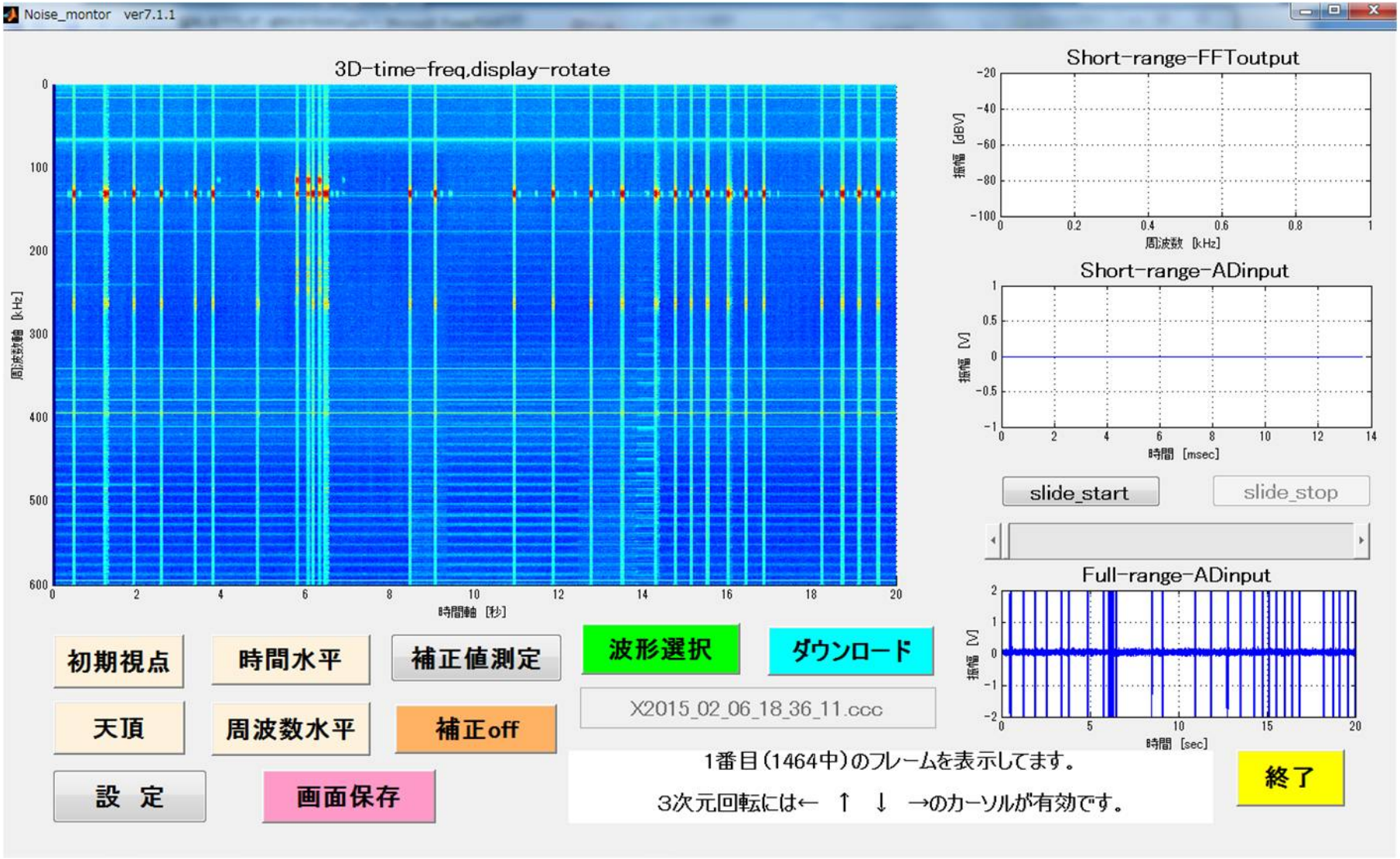



Elliptic

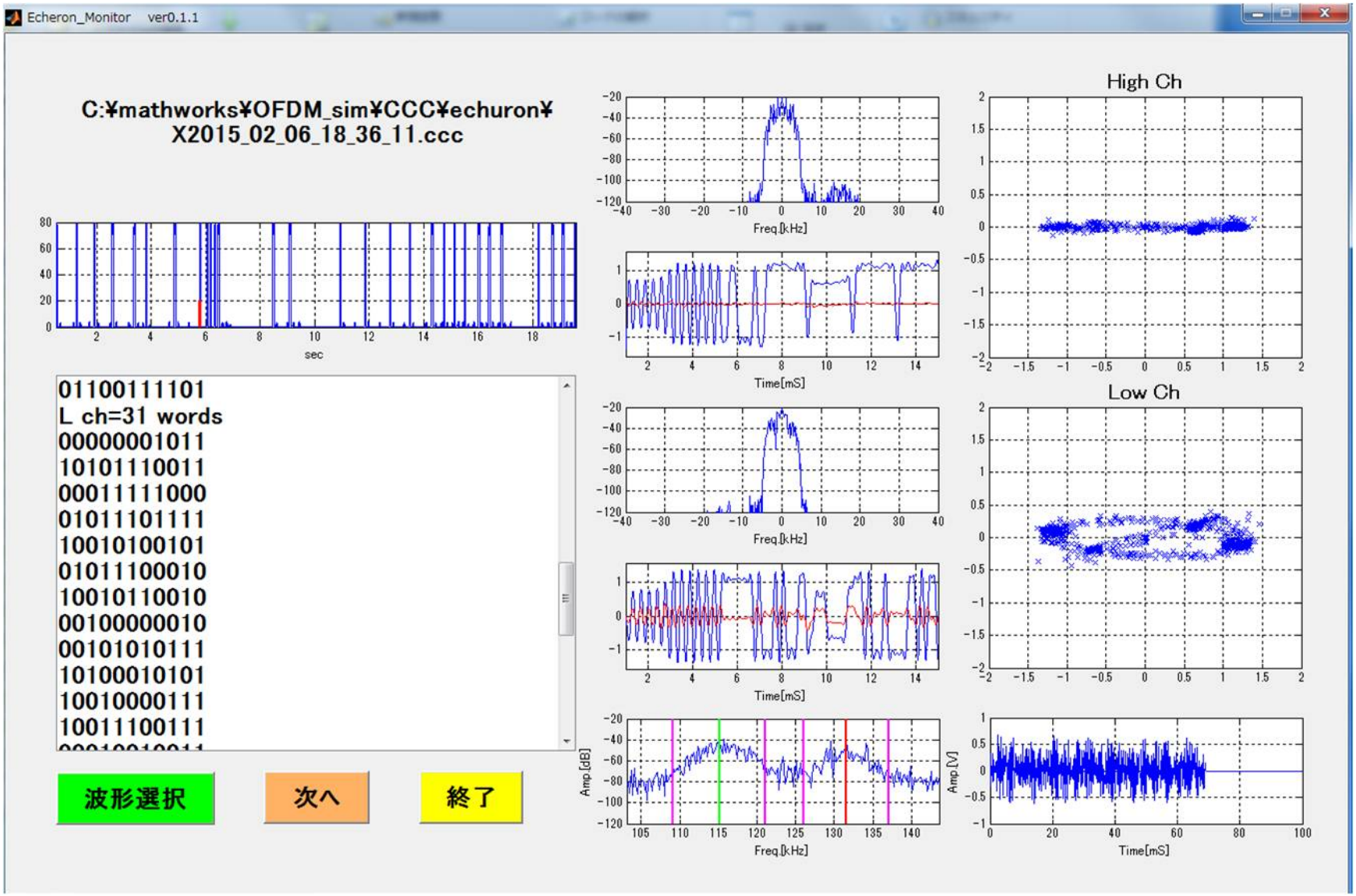
# MATLABで記述したＧＵＩの例ー１



# MATLABで記述したＧＵＩの例ー２



# MATLABで記述したＧＵＩの例ー３



# MATLAB-to-Cソース変換ー1

MATLAB

```
function [x1,y1,preg]=precoder_f(u1,c1,p1,preg)
y1=u1+c1;
x1=y1-p1;
preg(3)=real(preg(2))+imag(preg(2))*1i;
preg(2)=real(preg(1))+imag(preg(1))*1i;
preg(1)=real(x1)+imag(x1)*1i;
return
```

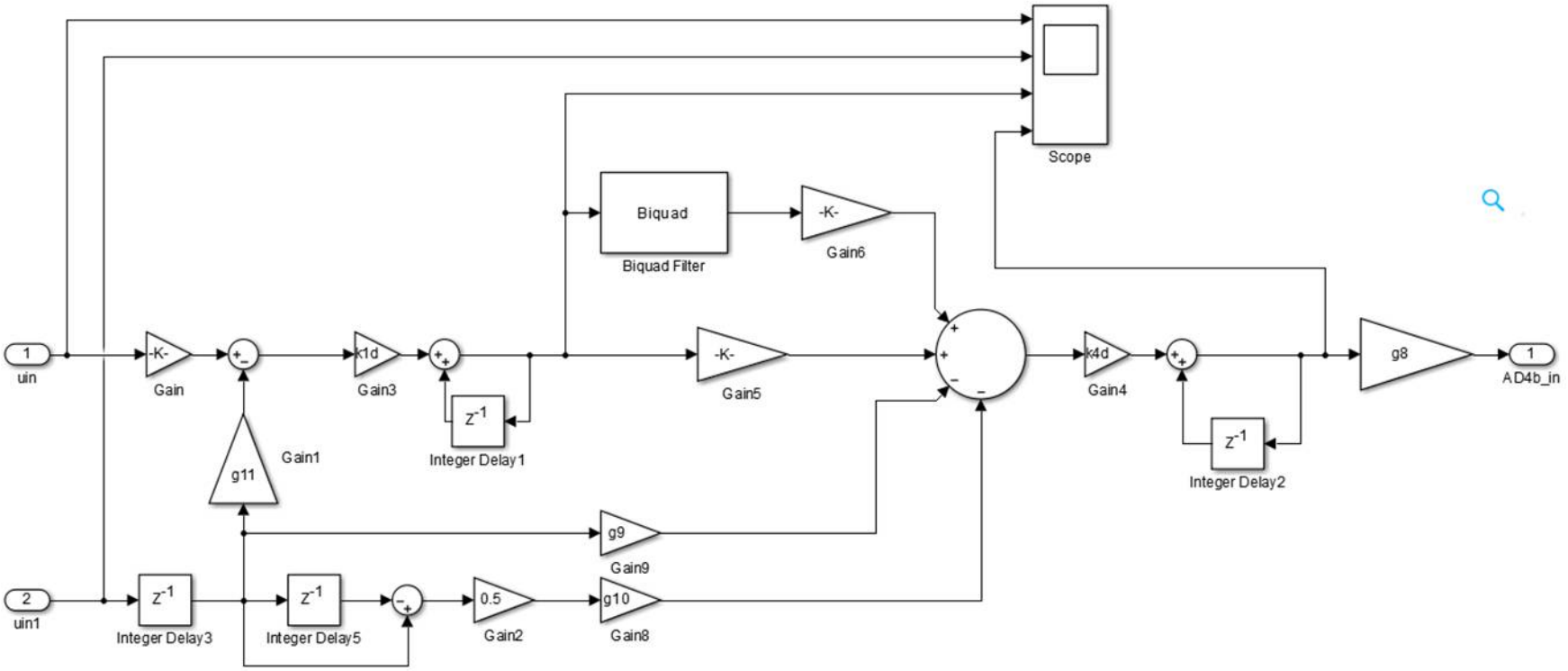
次頁　Cソース

# MATLAB-to-Cソース変換ー2

## Cソース

```
/* 'map_4D:33' [x1,y1,pregTX]=precoder_f(u1,c1,p1,pregTX); */
  /* global v34map */
  /* case int32(1) */
  /* 'precoder_f:4' y1=u1+c1; */
  d_y1->re = u1.re + c1.re;
  d_y1->im = u1.im + c1.im;
  /* 'precoder_f:5' x1=y1-p1; */
  c_x1->re = d_y1->re - p1.re;
  c_x1->im = d_y1->im - p1.im;
  /* 'precoder_f:6' preg(3)=preg(2); */
  b_pregTX[2] = pregTX[1];
  /* 'precoder_f:7' preg(2)=preg(1); */
  b_pregTX[1] = pregTX[0];
  /* 'precoder_f:8' preg(1)=x1; */
  b_pregTX[0] = *c_x1;
  for (i_Z1 = 0; i_Z1 < 3; i_Z1++) {
    pregTX[i_Z1] = b_pregTX[i_Z1];
  }
```

# simulink-to-HDL変換ー1



# simulink-to-HDL変換ー2

```
`timescale 1 ns / 1 ns
module SD4_new_filter_block1
          ( clk,reset, enb_1_20_0, uin, uin1,AD4b_in);
  input   clk;
  input   reset;
  input   enb_1_20_0;
  input   [63:0] uin;  // double
  input   [63:0] uin1;  // double
  output  [63:0] AD4b_in;  // double
                  ||
  wire [63:0] Sum1_out1_1;  // ufix64
  wire [63:0] Biquad_Filter_out1;  // ufix64
  real Biquad_Filter_out1_double;  // double
  real Gain6_out1;  // double
  real Gain5_out1;  // double
  real Sum3_op_stage2;  // double
```

# simulink-to-HDL変換ー3

```
always @* uin_double = $bitstoreal(uin);
// <S4>/Gain
always @* Gain_out1 = 0.001 * uin_double;
always @* uin1_double = $bitstoreal(uin1);
// <S4>/Integer Delay3
always @(posedge clk or posedge reset)
  begin : Integer_Delay3_process
    if (reset == 1'b1) begin
      Integer_Delay3_out1 <= 0.0;
    end
    else begin
      if (enb_1_20_0) begin
        Integer_Delay3_out1 <= uin1_double;
      end
    end
  end
```

# simulink-to-HDL変換ー4

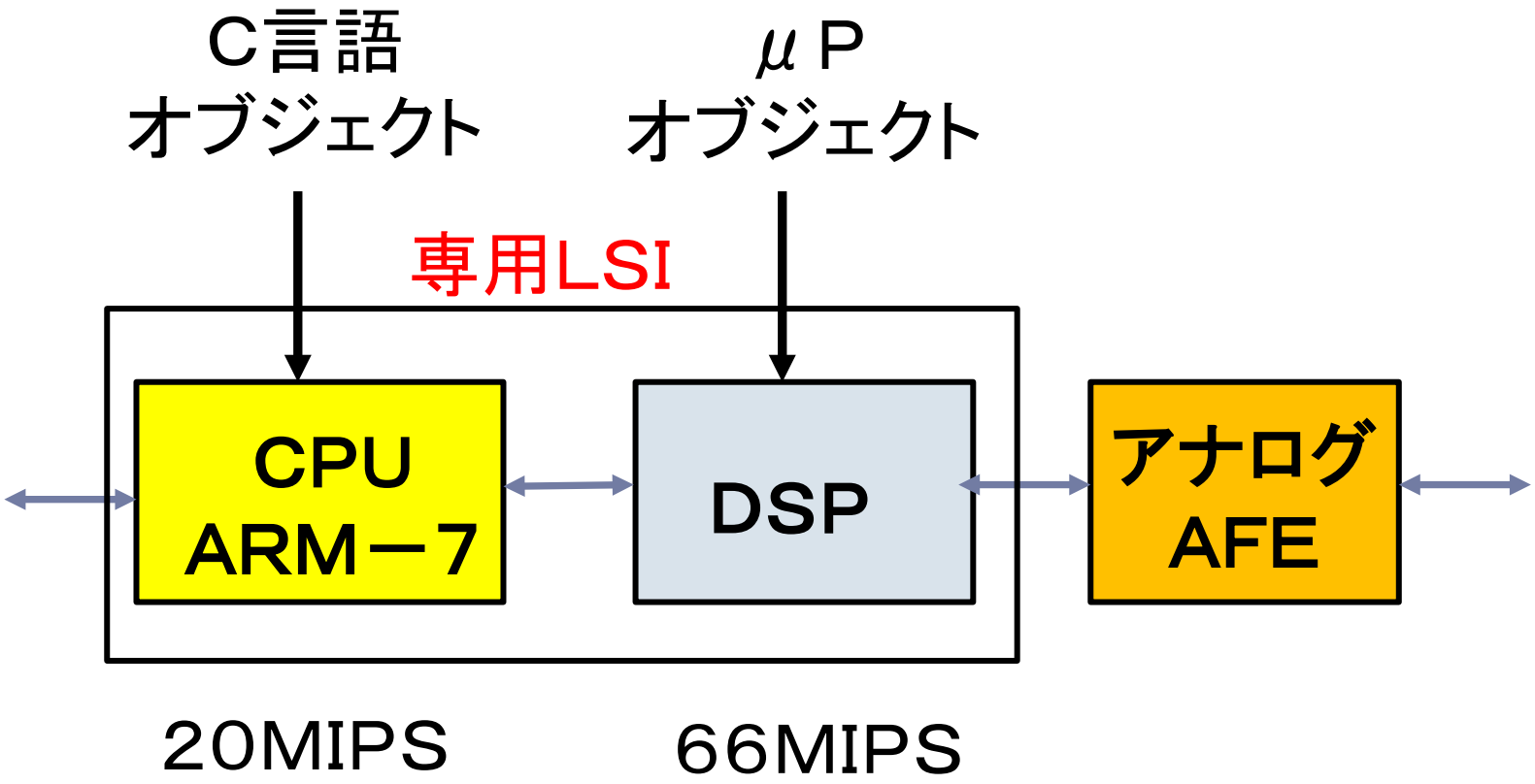
```
            ||
 // <S4>/Sum1
always @* Sum1_out1 = Gain3_out1 + Integer_Delay1_out1;
assign Sum1_out1_1 = $realtobits(Sum1_out1);
// <S4>/Biquad Filter
Biquad_Filter   u_Biquad_Filter   (.clk(clk),
                                   .enb_1_20_0(enb_1_20_0),
                                   .reset(reset),
                                   .Biquad_Filter_in(Sum1_out1_1),  // double
                                   .Biquad_Filter_out(Biquad_Filter_out1)  // double
                                   );
always @* Biquad_Filter_out1_double = $bitstoreal(Biquad_Filter_out1);
// <S4>/Gain6
always @* Gain6_out1 = 0.0016666666666666668 * Biquad_Filter_out1_double;
// <S4>/Gain5
always @* Gain5_out1 = 0.0033333333333333335 * Sum1_out1;
always @* Sum3_op_stage2 = Gain6_out1 + Gain5_out1;
```

# ｓｉｍｕｌｉｎｋ-to-ＨＤＬ変換ー5

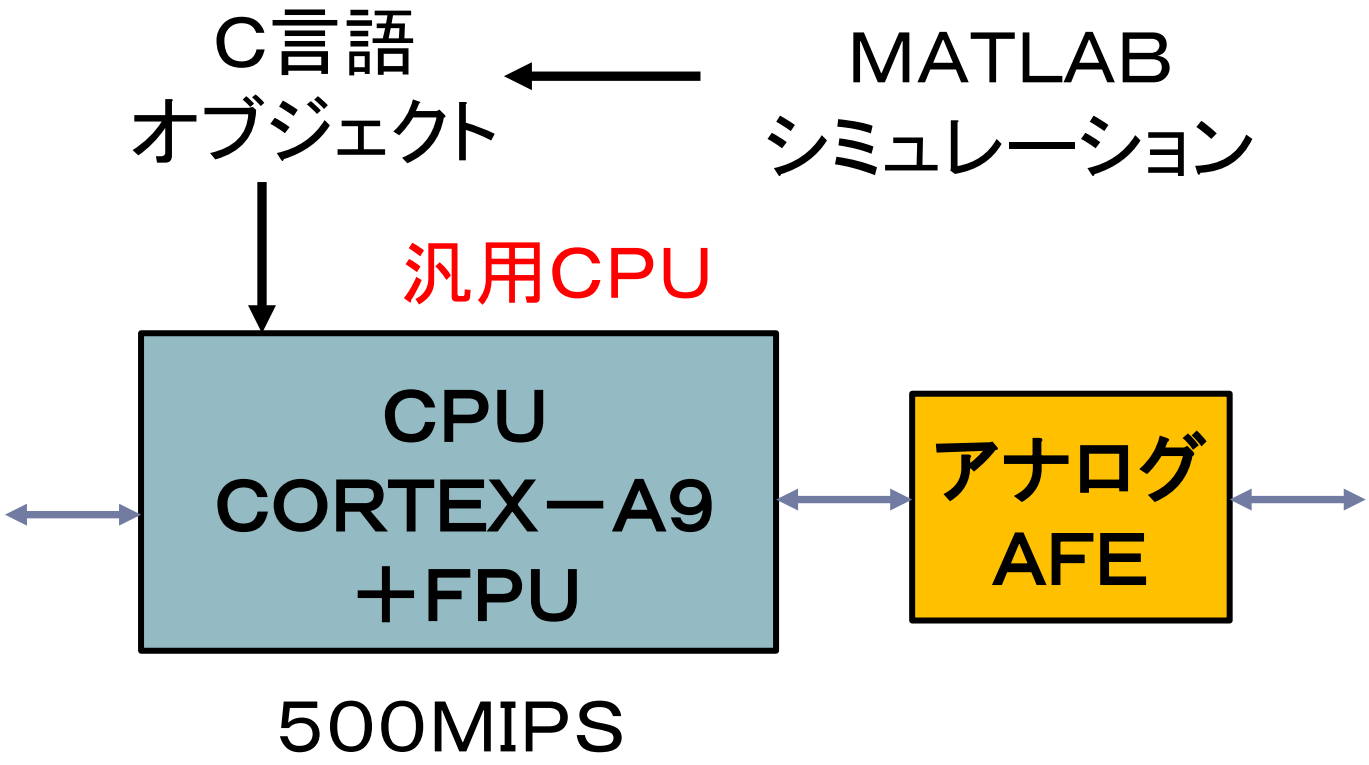
```
                    ||
  // <S4>/Sum2
  always @* Sum2_out1 = Gain4_out1 + Integer_Delay2_out1;
  // <S4>/c8
  always @* c8_out1 = 8.0 * Sum2_out1;
  assign AD4b_in = $realtobits(c8_out1);
  // <S4>/Scope
endmodule  // SD4_new_filter_block1
```

# 6. CPU速度の進化

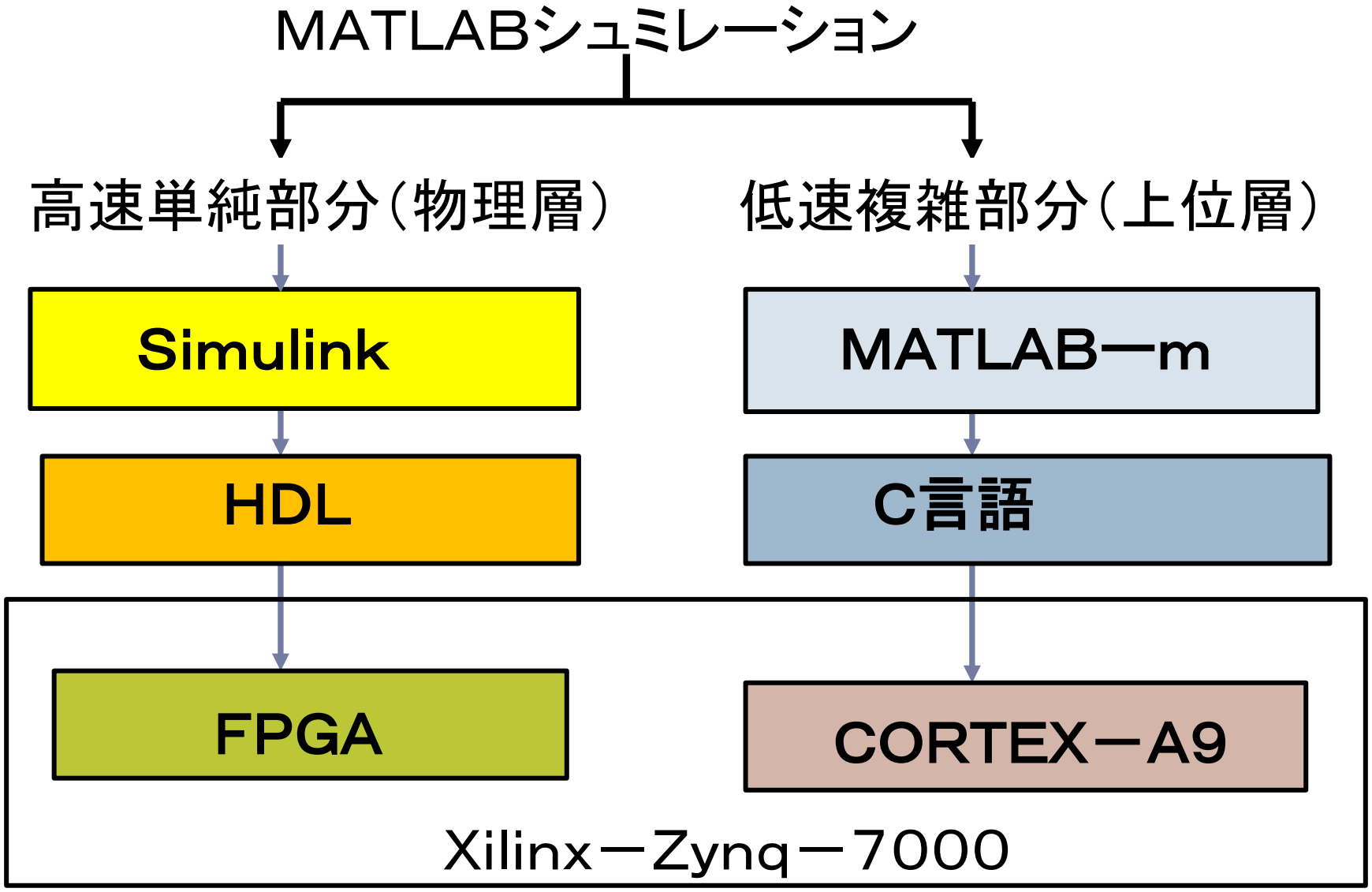
# ＭＡＴＬＡＢ以前のモデム開発



# ＭＡＴＬＡＢ以後のモデム開発



# 将来の通信ＬＳＩ開発の手法



# Xilinx-Zynq-7000 その 1

## 機能一覧

表 1 : Zynq-7000 All Programmable SoC

| Zynq-7000 All Programmable SoC | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | デバイス名 | Z-7010 | Z-7015 | Z-7020 | Z-7030 | Z-7035 | Z-7045 | Z-7100 |
| | デバイス番号 | XC7Z010 | XC7Z015 | XC7Z020 | XC7Z030 | XC7Z035 | XC7Z045 | XC7Z100 |
| プロセッシング システム | プロセッサ コア | CoreSight™ を搭載したデュアル ARM® Cortex™-A9 MPCore™ | | | | | | |
| | プロセッサの拡張機能 | 各プロセッサに NEON™ および単精度/倍精度浮動小数点ユニット | | | | | | |
| | 最大周波数 | 667MHz (-1); 766MHz (-2); 866MHz (-3) | | | 667MHz (-1); 800MHz (-2); 1GHz (-3) | | | 667MHz (-1); 800MHz (-2) |
| | L1 キャッシュ | 各プロセッサに 32KB 命令キャッシュと 32KB データ キャッシュ | | | | | | |
| | L2 キャッシュ | 512KB | | | | | | |
| | オンチップ メモリ | 256KB | | | | | | |
| | 外部メモリ サポート(1) | DDR3、DDR3L、DDR2、LPDDR2 | | | | | | |
| | 外部スタティック メモリ サポート(1) | クワッド SPI x2、NAND、NOR | | | | | | |
| | DMA チャネル | 8 (4 つはプログラマブル ロジック専用) | | | | | | |
| | ペリフェラル(1) | UART x2、CAN 2.0B x2、I2C x2、SPI x2、32b GPIO x4 | | | | | | |
| | DMA 内蔵ペリフェラル(1) | USB 2.0 (OTG) x2、トライモード ギガビット イーサネット x2、SD/SDIO x2 | | | | | | |
| | セキュリティ (2) | RSA 認証、256 ビットの AES および SHA 複合/認証によるセキュア ブート | | | | | | |

# Xilinx-Zynq-7000 その２



表 1 : Zynq-7000 All Programmable SoC (続き)

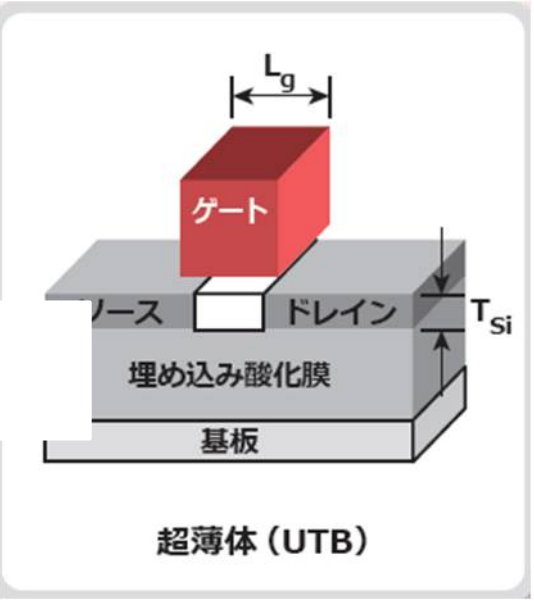
| Zynq-7000 All Programmable SoC | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | デバイス名 | Z-7010 | Z-7015 | Z-7020 | Z-7030 | Z-7035 | Z-7045 | Z-7100 |
| | デバイス番号 | XC7Z010 | XC7Z015 | XC7Z020 | XC7Z030 | XC7Z035 | XC7Z045 | XC7Z100 |
| プロセッシング システムと プログラマブル ロジックの インターフェイス ポート (プライマリ インターフェイスおよび 割り込みのみ) | | AXI 32 ビット マスター x2、AXI 32 ビット スレーブ x2<br>AXI 64 ビット/32 ビット メモリ x4<br>AXI 64 ビット ACP<br>16 個の割り込み | | | | | | |
| プログラマブル ロジック | 相当するザイリンクス 7 シリーズ プログラマブル ロジック | Artix®-7 FPGA | Artix-7 FPGA | Artix-7 FPGA | Kintex®-7 FPGA | Kintex-7 FPGA | Kintex-7 FPGA | Kintex-7 FPGA |
| | プログラマブル ロジック セル (ASIC ゲート相当数)(3) | 28K ロジック セル (~430K) | 74K ロジック セル (~1.1M) | 85K ロジック セル (~1.3M) | 125K ロジック セル (~1.9M) | 275K ロジック セル (~4.1M) | 350K ロジック セル (~5.2M) | 444K ロジック セル (~6.6M) |
| | ルックアップ テーブル (LUT) | 17,600 | 46,200 | 53,200 | 78,600 | 171,900 | 218,600 | 277,400 |
| | フリップフロップ | 35,200 | 92,400 | 106,400 | 157,200 | 343,800 | 437,200 | 554,800 |
| | エクステンシブル ブロック RAM (36Kb ブロックの数) | 240KB (60) | 380KB (95) | 560KB (140) | 1,060KB (265) | 2,000KB (500) | 2,180KB (545) | 3,020KB (755) |
| | プログラマブル DSP スライス (18 × 25 MACC) | 80 | 160 | 220 | 400 | 900 | 900 | 2,020 |
| | DSP の最大処理速度 (対称 FIR) | 100GMAC | 200GMAC | 276GMAC | 593GMAC | 1,334GMAC | 1,334GMAC | 2,622GMAC |
| | PCI Express® (ルート コンプレックス またはエンドポイント)(4) | — | Gen2 x4 | — | Gen2 x4 | Gen2 x8 | Gen2 x8 | Gen2 x8 |
| | アナログ ミックスド シグナル (AMS)/ XADC | 最大 17 の差動入力を備えた 12 ビット 1MSPS ADC x2 | | | | | | |
| | セキュリティ (2) | AES および SHA 256b によるブート コードおよび PL のコンフィギュレーション、復号、認証 | | | | | | |

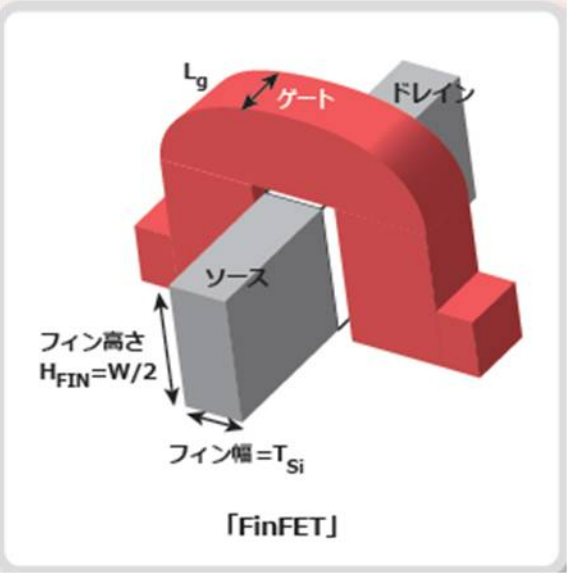
＝カスタムLSIはもはや必要が無い！！

# 「ムーアの法則」の終えん ？？

LSIの微細化は、加工技術やデバイス動作の物理限界により、20nm程度で頭打ちになると予測され、10年程度で「ムーアの法則」は終えんを迎えると考えられている。

# ＦｉｎＦＥＴが時代をブレーク



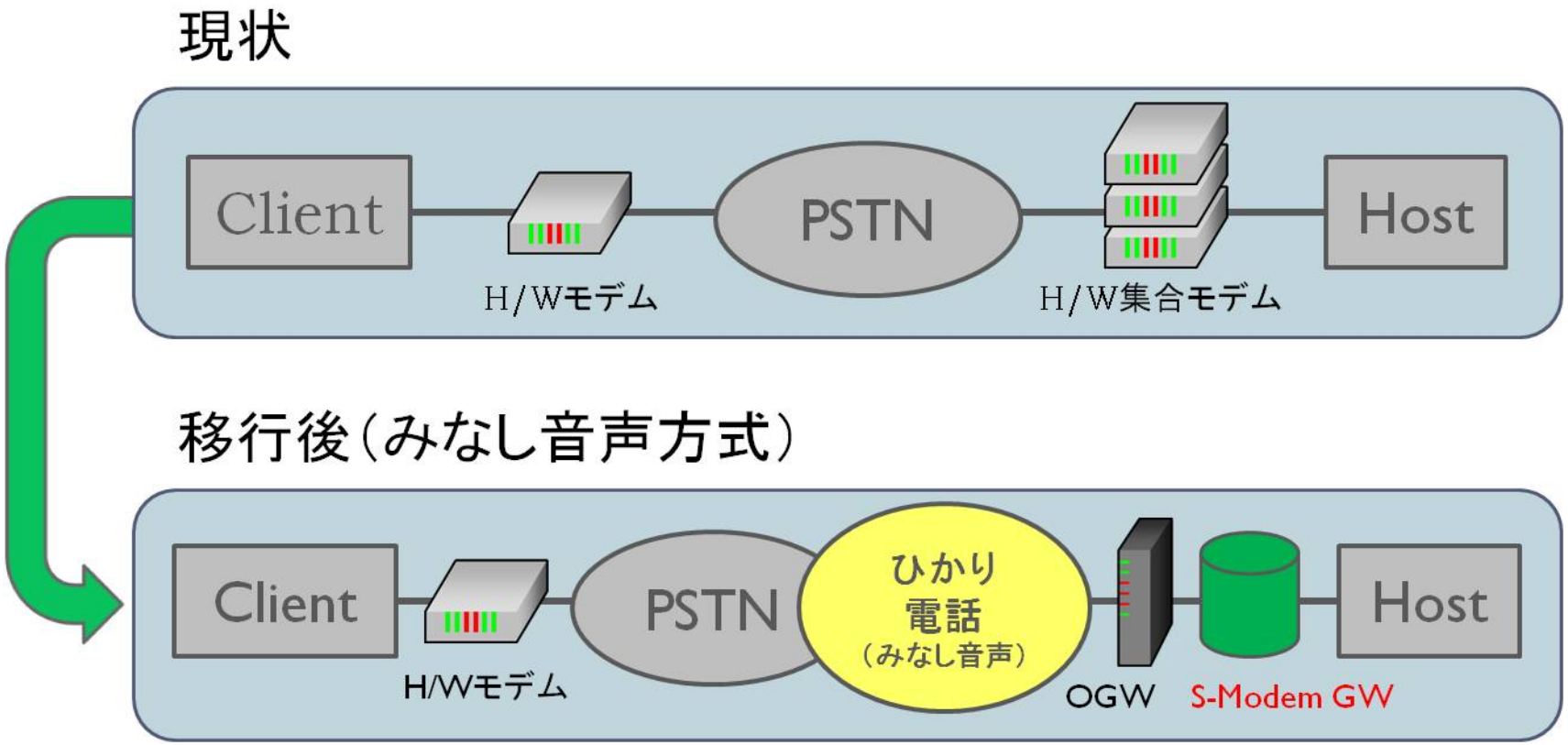

ファウンダリ各社の16/14nm 3DトランジスタFinFETプロセスは、20nmプロセスに対してパフォーマンスロジックチップのサイズが縮小する。これは、FinFETプロセスで、スタンダードセルが大きく変わるからだ。特に高性能チップ用のセルライブラリは、セルのサイズが小さくなり、より小さいサイズで高性能を達成できるようになる。

# 7. ハードモデムからソフトモデムへ

# ソフトモデム移行への必然性

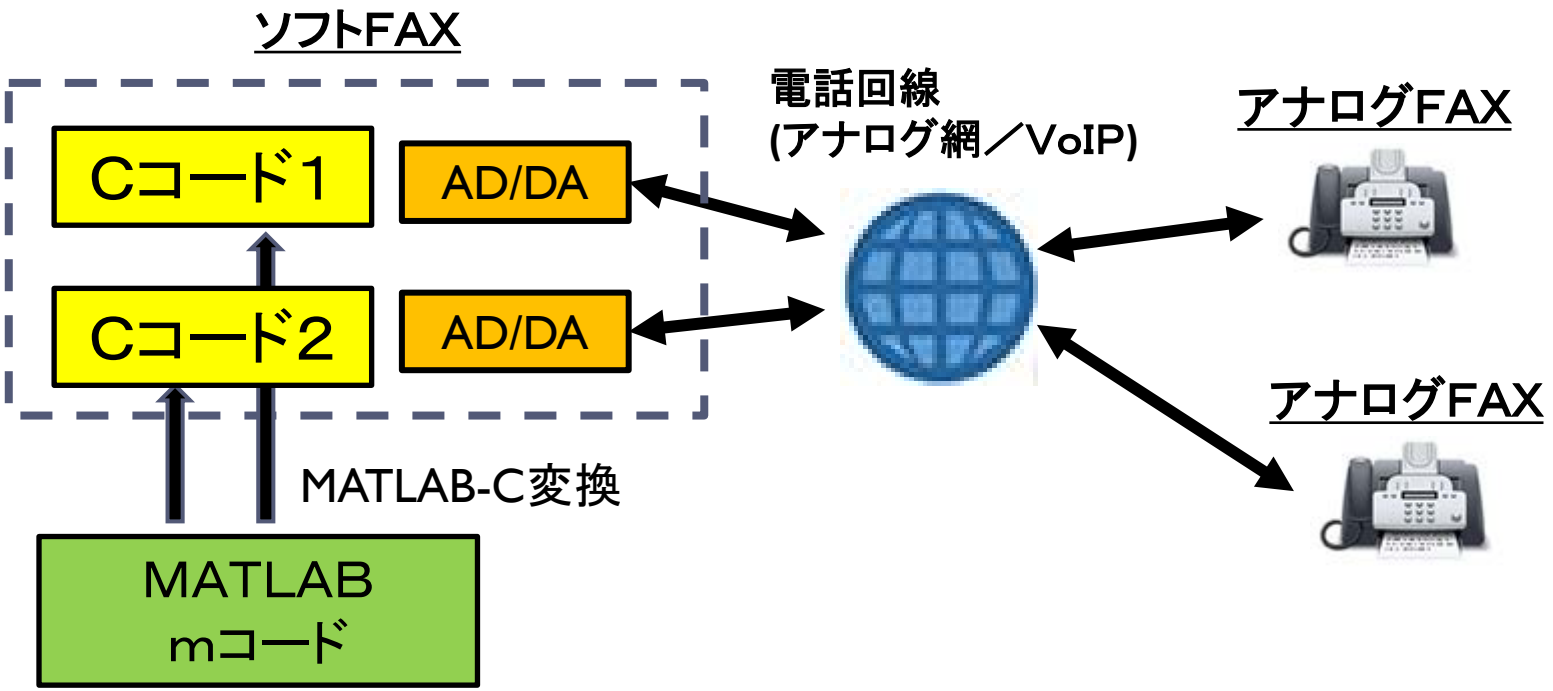
1. DSPはすでに、新規開発されていない。
2. 専用LSI（ASSP）のレイアウトルール縮小
   - 130nmから30nmへの移行により、
     ＝＝＝＞130nm拡散工場の消滅
   - 30nmマスク枚数が増加し、開発費の増加、約2億円
   - 検証作業が複雑化
     ＝＝＝＞ASSPの生産打ち切り（ディスコン）
3. チップ供給は、年間200万個を切ると、供給されない。
   ＝＝＝＞汎用CPUによるソフトモデムへ移行

# センターサーバーのソフトモデム化



# 複数チャンネル対応可能

・ＡＤ／ＤＡ１個で、１チャンネルのモデム実装可能
・Ｖ３４送受信は１２０ＭＦＲＯＰＳで実行可能、
　　１ＧＨｚ－ＣＰＵ＝８チャンネル対応



ご清聴ありがとうございました。

http://www.egretcom.com