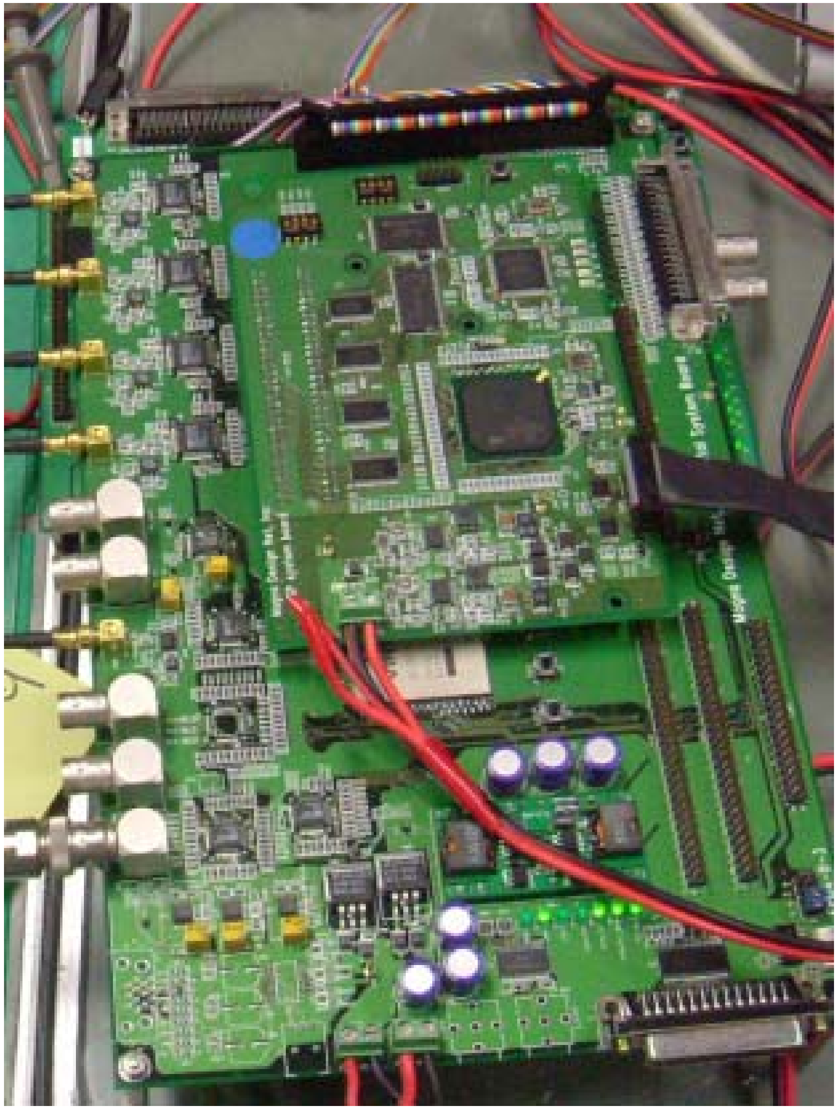
Implementation Example - DSP based Adaptive Array Antenna System -

Fire Tom Wada

Professor, Information Engineering, Univ. of the Ryukyus

DSP based Adaptive Array Antenna System

- DSP based AAA System for OFDM receiver is shown as a implementation example.
- The System is composed of three parts.
  1. OFDM demodulator
  2. Adaptive Array Antenna
  3. DSP

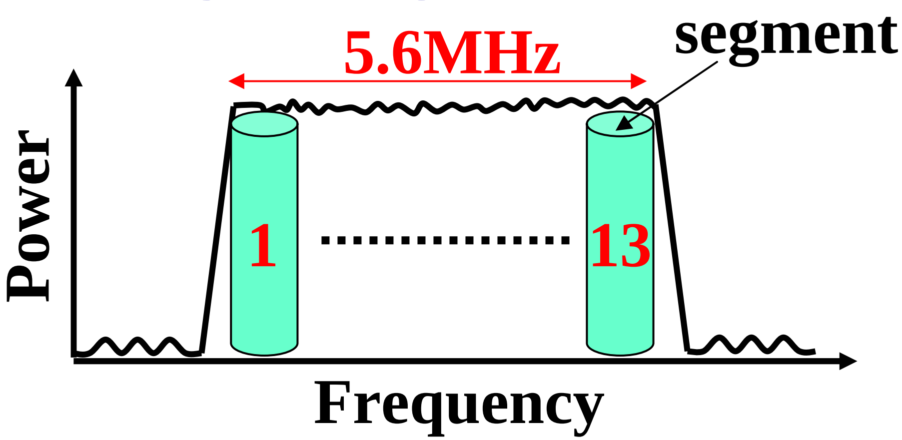


OUTLINE

1. ISDB-T abstract
2. OFDM demodulator
3. Adaptive Array Antenna System
4. System Design

Terrestrial Digital TV in Japan

- **BST-OFDM**



Modulation:

64QAM, 16QAM, QPSK

Number of sub-carrier

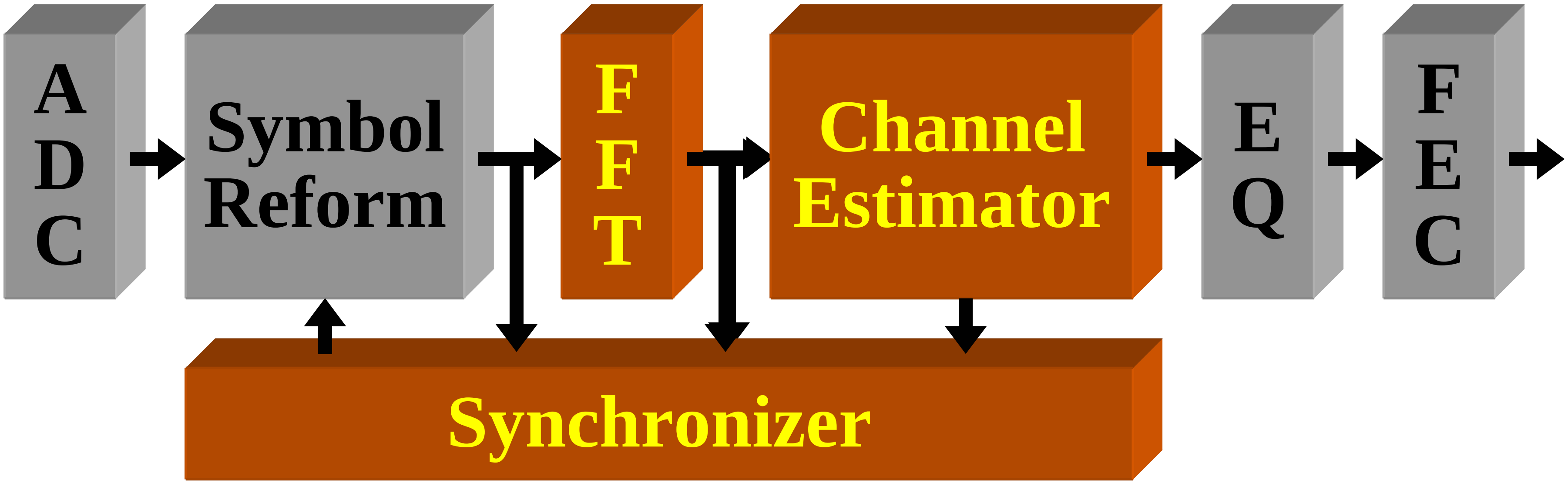
192(Mode2) / 384(Mode3)

- **The feature of OFDM**
  - **Long symbol duration composed by sub-carriers with a guard time**
    - **Inter-symbol interference is eliminated**
    - **Multi-path distortion to be reduced**

Today's Broadcast (ISDB-T)

Broadcast	HDTV	Handheld
Modulation	64QAM (13segment)	QPSK (1segment)
Data Rate	~15Mbps	~370Kbps
Availability	2003	2005/E
Usage	Home-use	Mobile
Quality/Mobility	High / Low	Low / High

Simplified OFDM Receiver Model



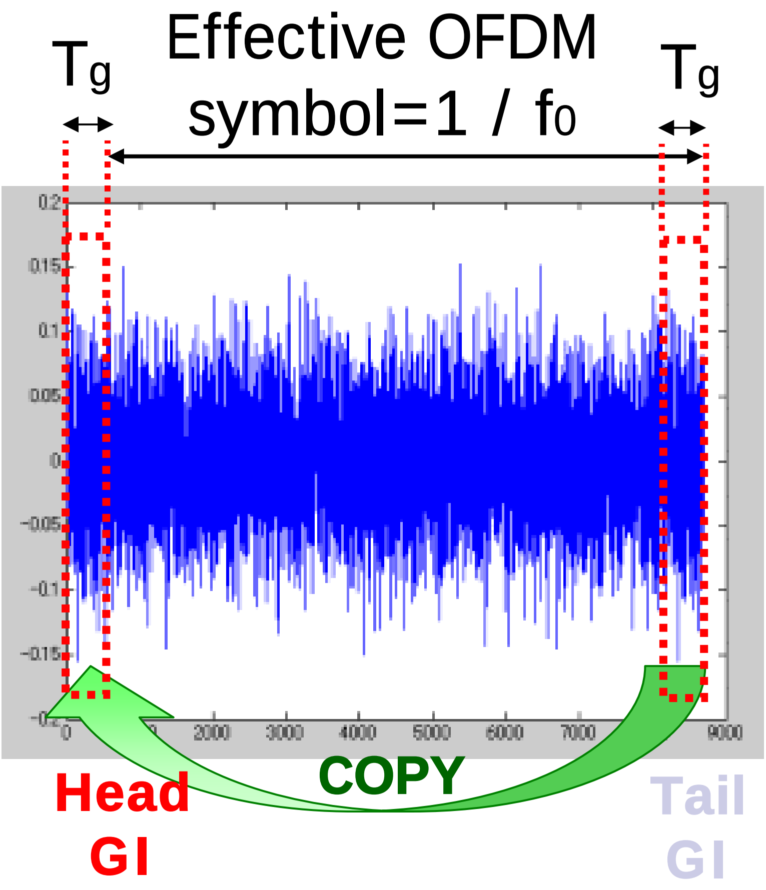
- Accurate and Agile Synchronizer
- Broad Dynamic Range of FFT
- Sophisticated Channel Estimation

Guard Interval of OFDM signal

- In order to prevent (n-1) delay symbol from interfering to n symbol,
  GI is pre-appended as a copy of the tail of the Effective OFDM symbol.
- We call Head-GI and Tail-GI.
- Head-GI and Tail-GI will be used in the AAA signal processing.

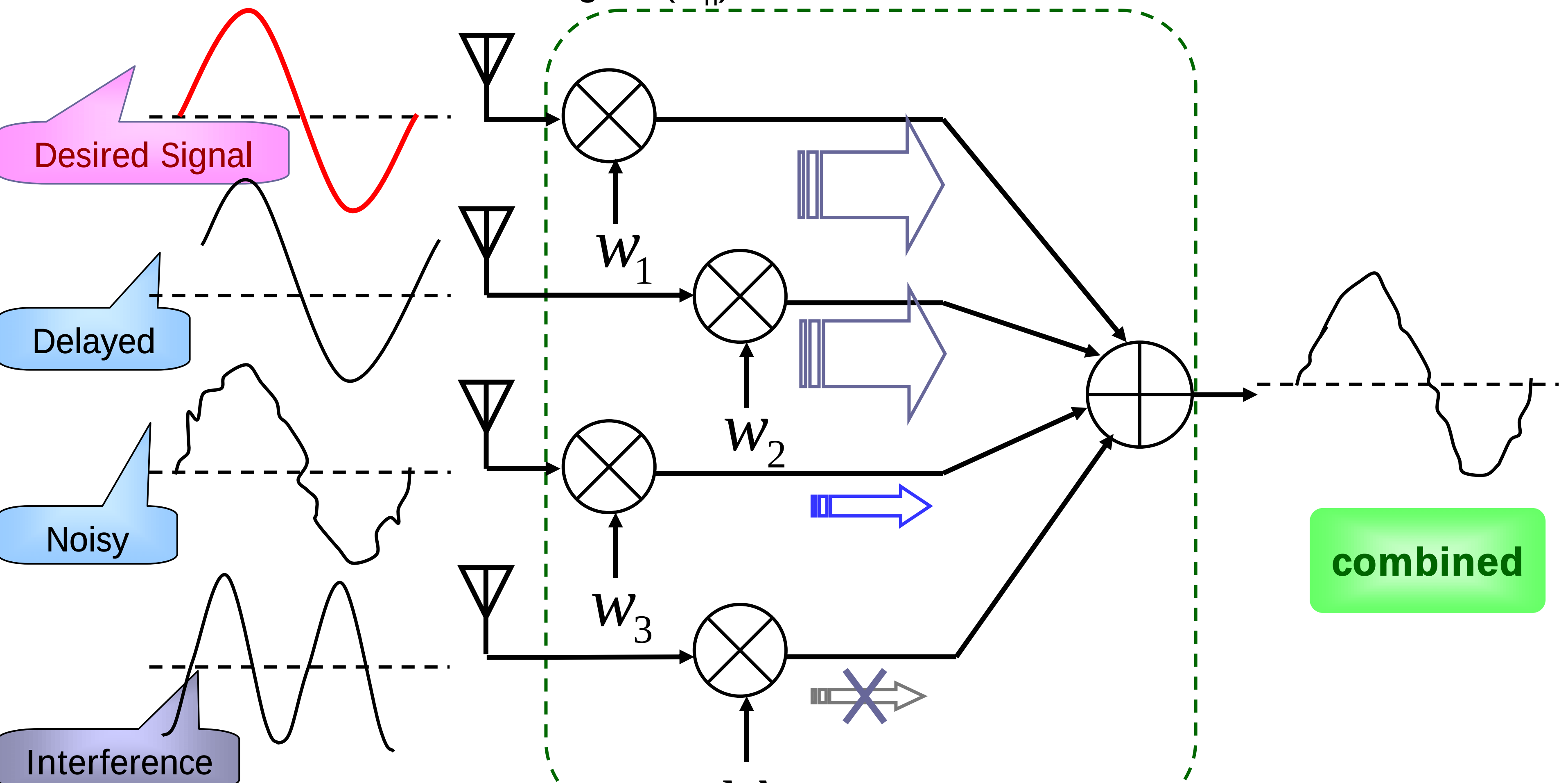
Data: 8K points
+ GI: 512 points

8704 points
Mode3:GI(1/16)

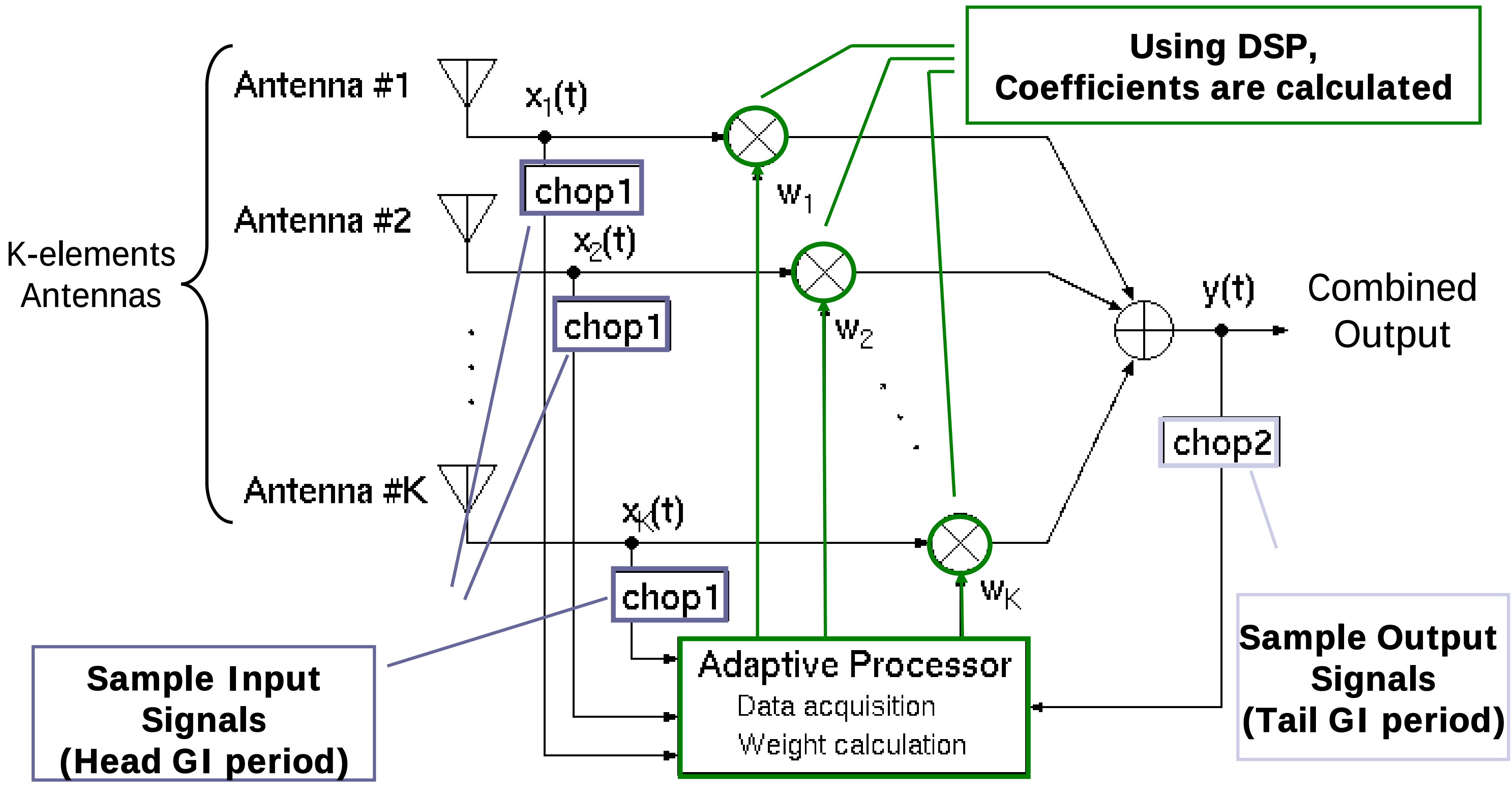


Adaptive Array Antenna

- Using multiple Antenna, signals are combined to reproduce a clean signal .
- Complex multiply and complex addition is used.
- DSP to calculate those weights (w_n).



AAA signal processing



Adaptive Algorithms

- Asynchronous
  1. Maximum Ratio Combining_Asyn
- Synchronous
  2. Maximum Ratio Combining_Syn
  3. Sample Matrix Inversion
  4. Power Inversion

	Wave	Adaptive Beam-forming	Adaptive Null Steering
1. MRC_ASYN	ANY	○	×
2. MRC_SYN	OFDM	○	×
3. SMI	OFDM	○	○
4. PI	OFDM	×	○

Adaptive Beam-forming
- Emphasize the desired Signal

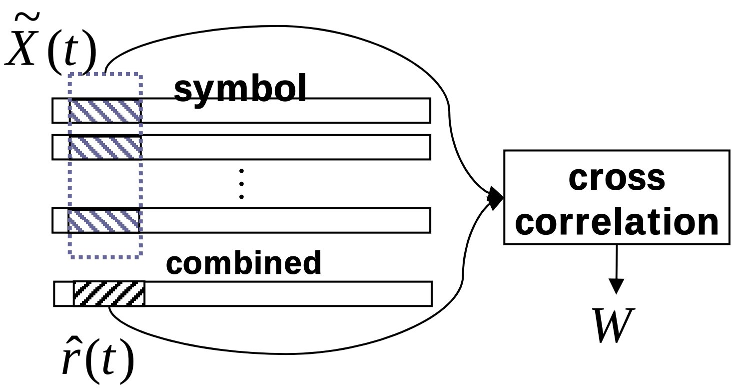
Adaptive Null Steering
- Suppress interference signal

Since the algorithm should be flexible, S/W approach is better!

MRC(Maximum ratio combining)

- Coefficients are calculated by cross-correlation of input signals and combined signal.
  - MRC_ASYN

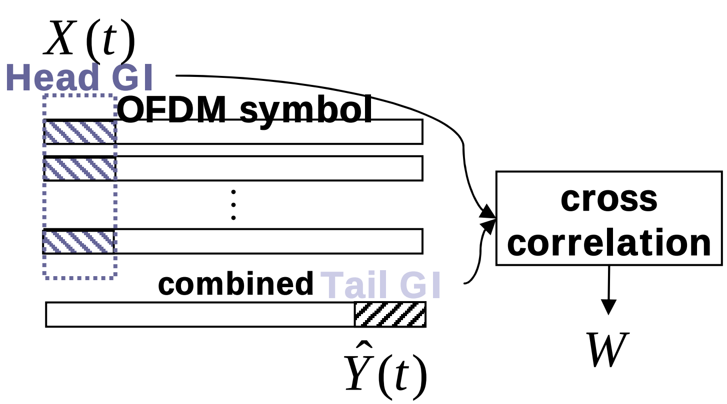
$$W = E[\tilde{X}(t) \cdot \hat{r}^*(t)]$$



  - MRC_SYN

    Head_GI = Tail_GI property is used.

$$W = E[X(t) \cdot \hat{Y}^*(t)]$$



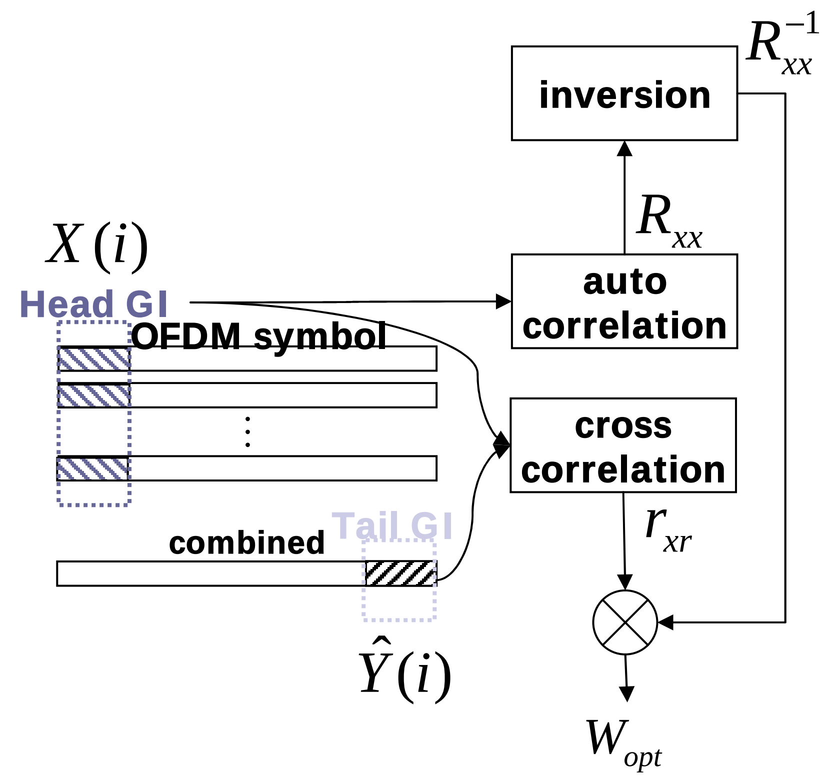
SMI(Sample Matrix Inversion)

- SMI needs reference signal
  - Here Head_G I= Tail_GI property is used.

$$r_{xr}(m) = \frac{1}{m}\sum_{i=1}^{m} X(i)\hat{Y}^*(i)$$

$$R_{xx}(m) = \frac{1}{m}\sum_{i=1}^{m} X(i)X^H(i)$$

$$W_{opt} = R_{xx}^{-1}(m) r_{xr}(m)$$



PI(Power Inversion)

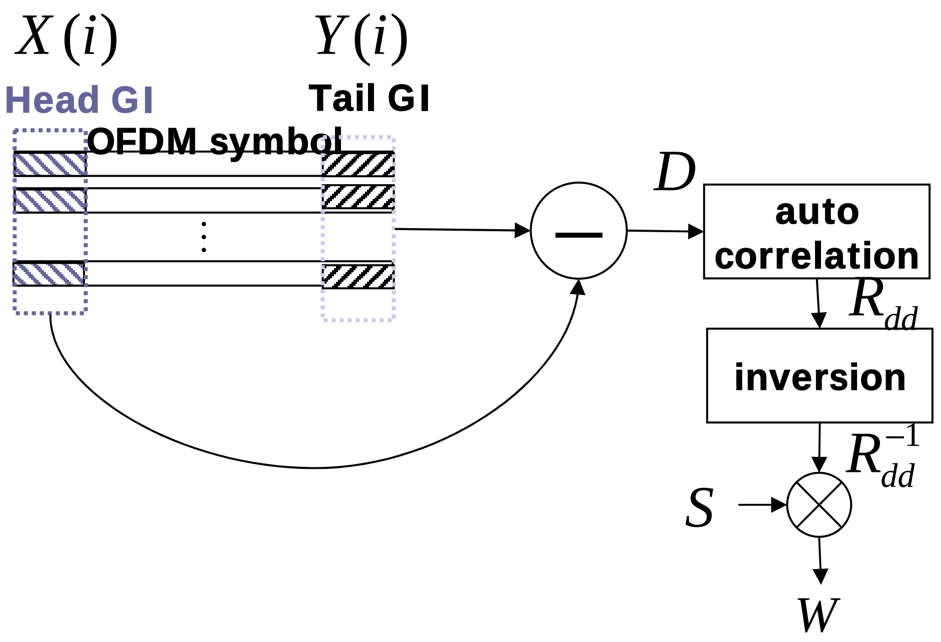
- PI algorithm can suppress maximum signal.
  =(Power Inversion)
- Here, we try to suppress the Difference of Head_GI and Tail_GI.

$$D = X(t) - Y(t)$$

$$R_{dd} = E\left[D \cdot D^{H}\right]$$

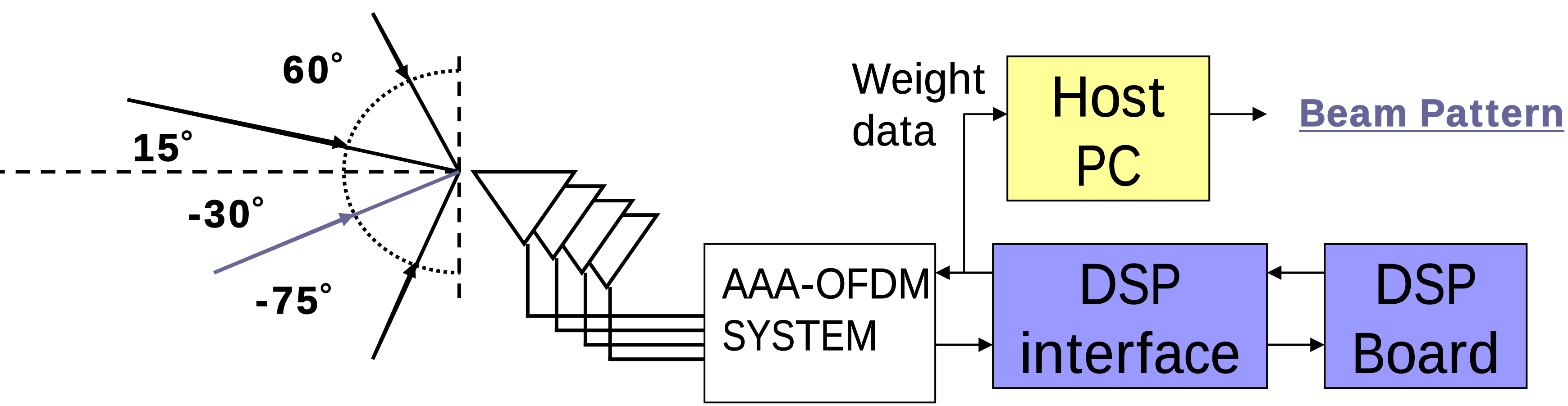
$$W = R_{dd}^{-1} \cdot S$$

$$S = [1,0,0,0]^{T}$$

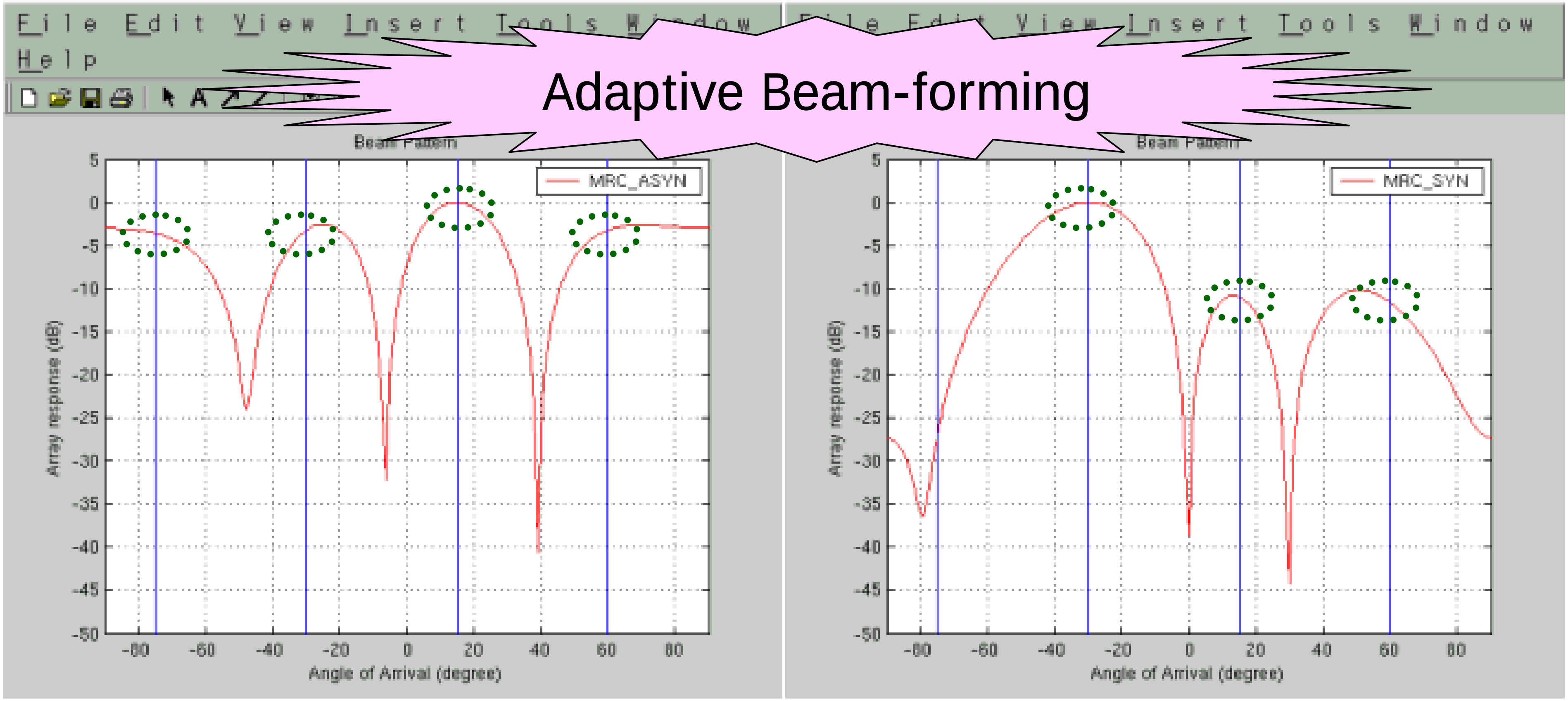


Evaluation Condition

		Base signal	Angle of Arrival	Delay [μs]	Power [dB]
Desired	Signal #1	DTV28CH	-30	0	0
Delay	Signal #2	DTV28CH	15	3/8 * T_g	0
Delay	Signal #3	DTV28CH	-75	6/8 * T_g	0
Interfe renece	Signal #4	DTV28CH	60	9/8 * T_g	0

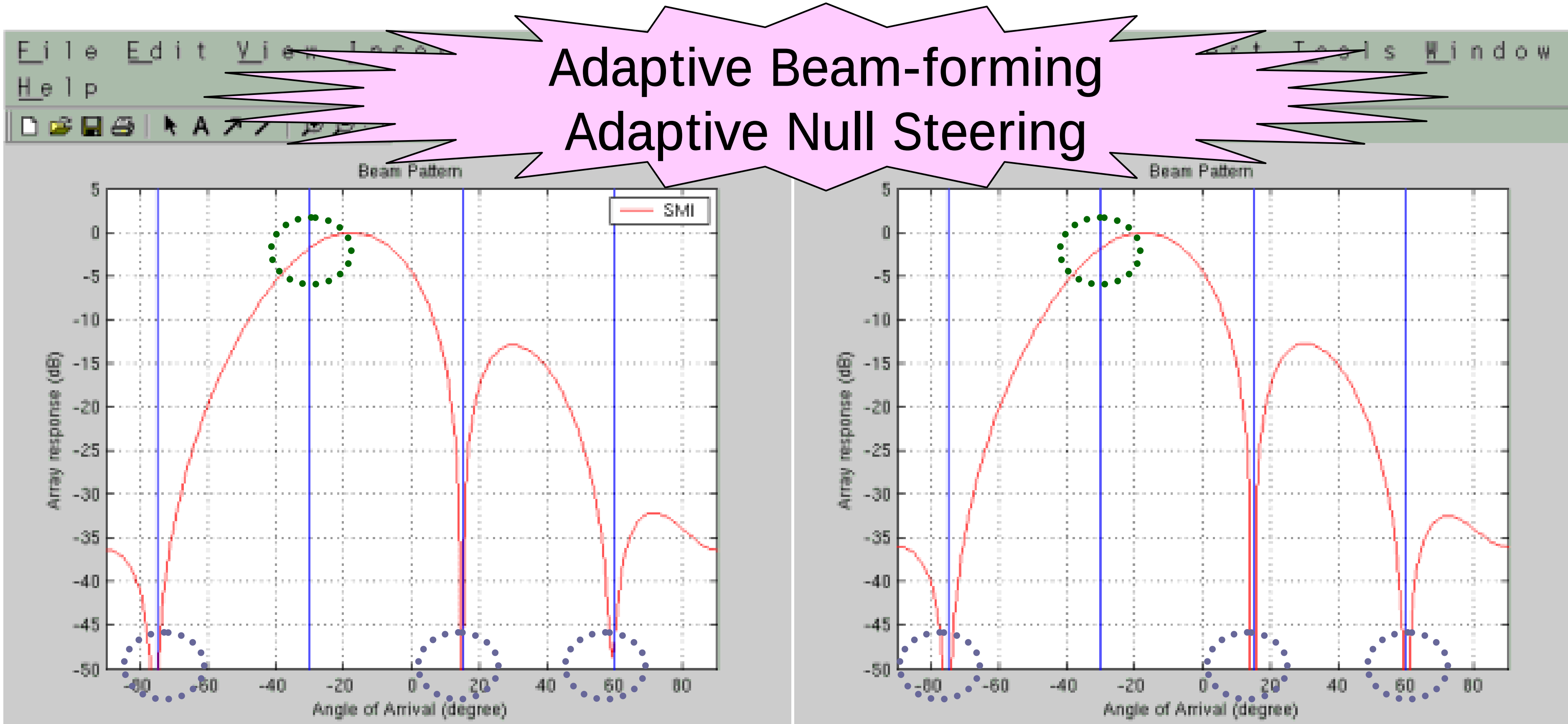


MATLAB Simulation [MRC_ASYN, MRC_SYN)]



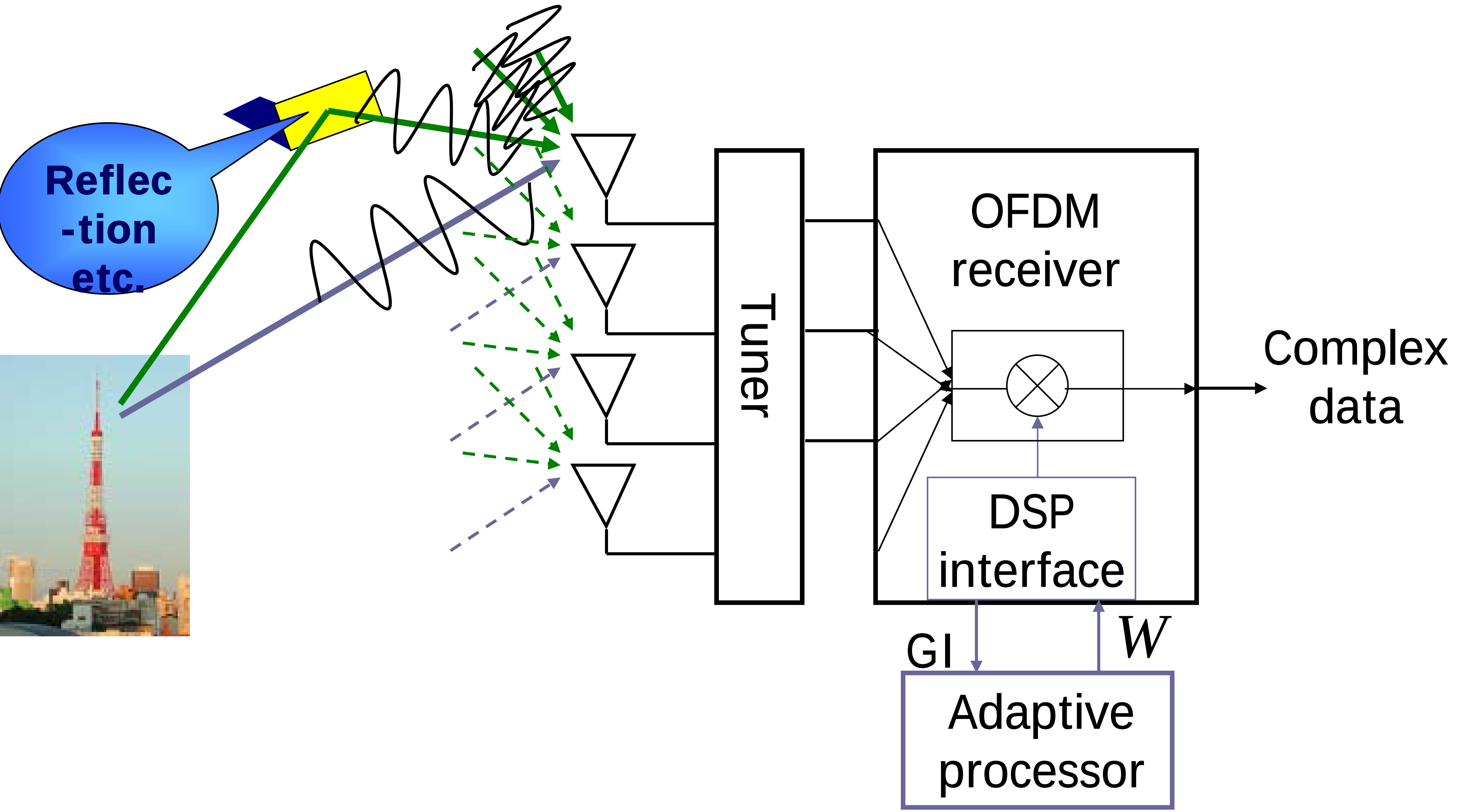
MRC_ASYN MRC_SYN

MATLAB Simulation [SMI,PI]

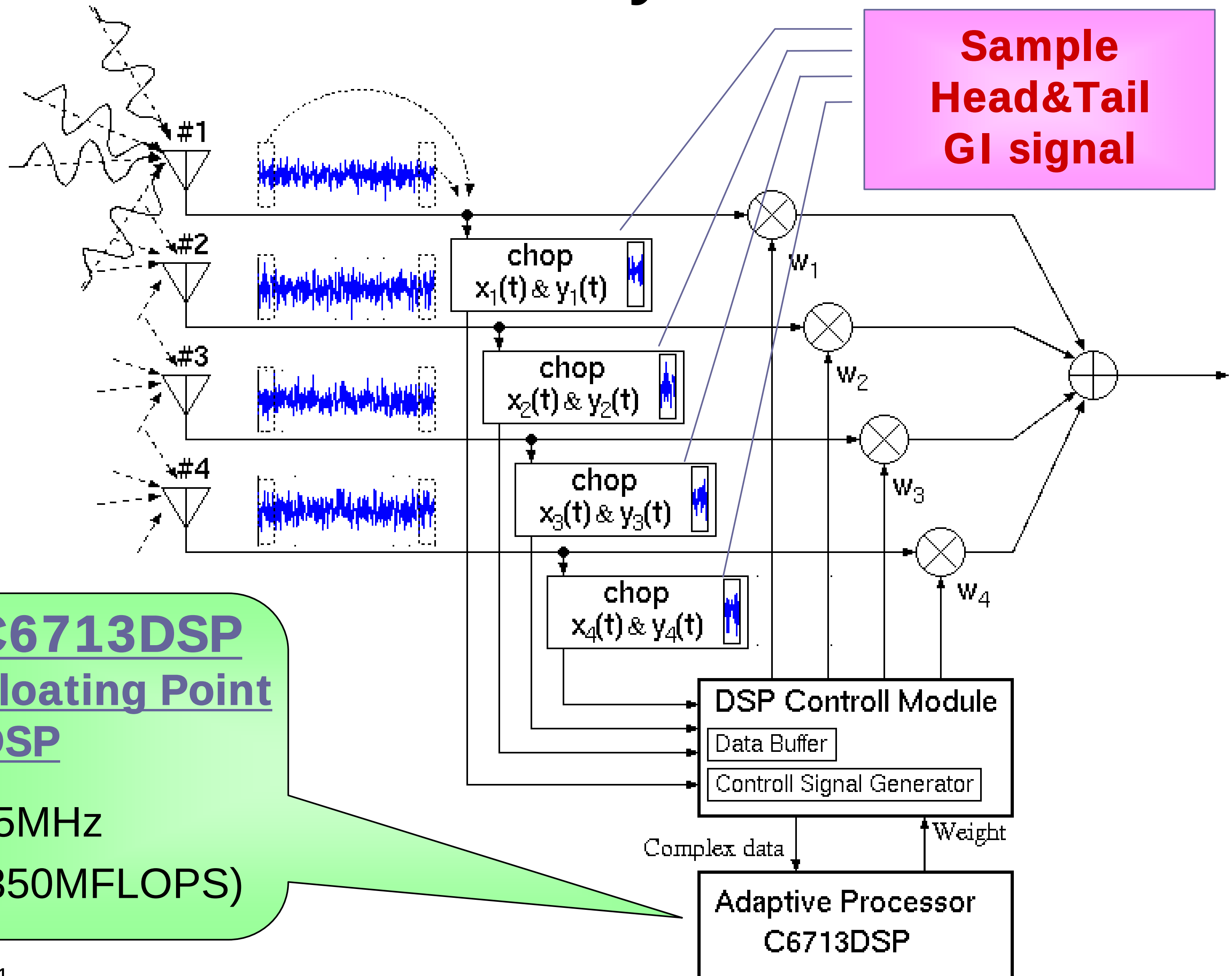


SMI　　　　PI

SYSTEM DESIGN



DSP based AAA System



C6713DSP
Floating Point DSP

225MHz
(1350MFLOPS)

TMS320C6713 DSP

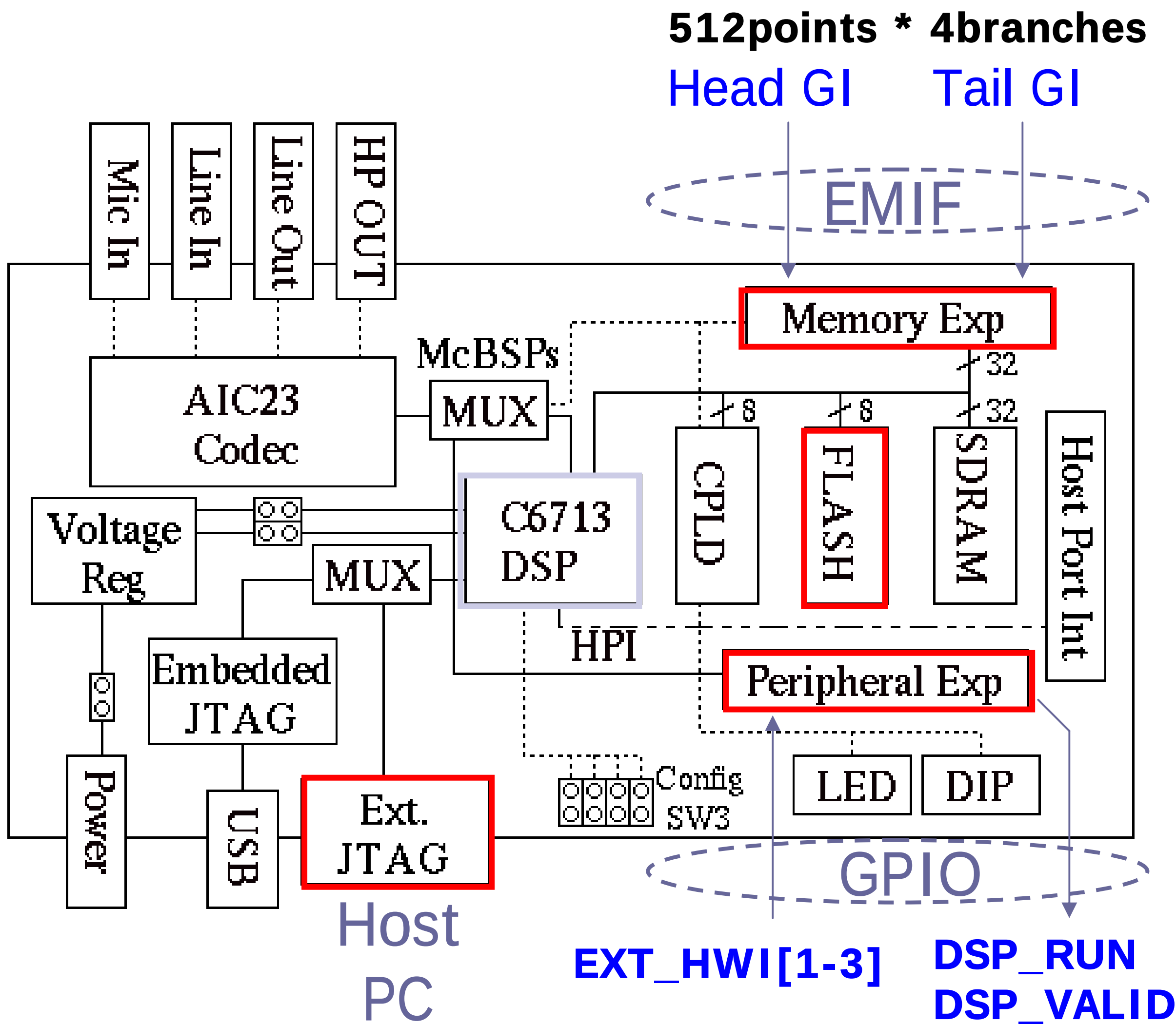
[Texas Instruments Inc, Floating point DSP]

C6713DSP

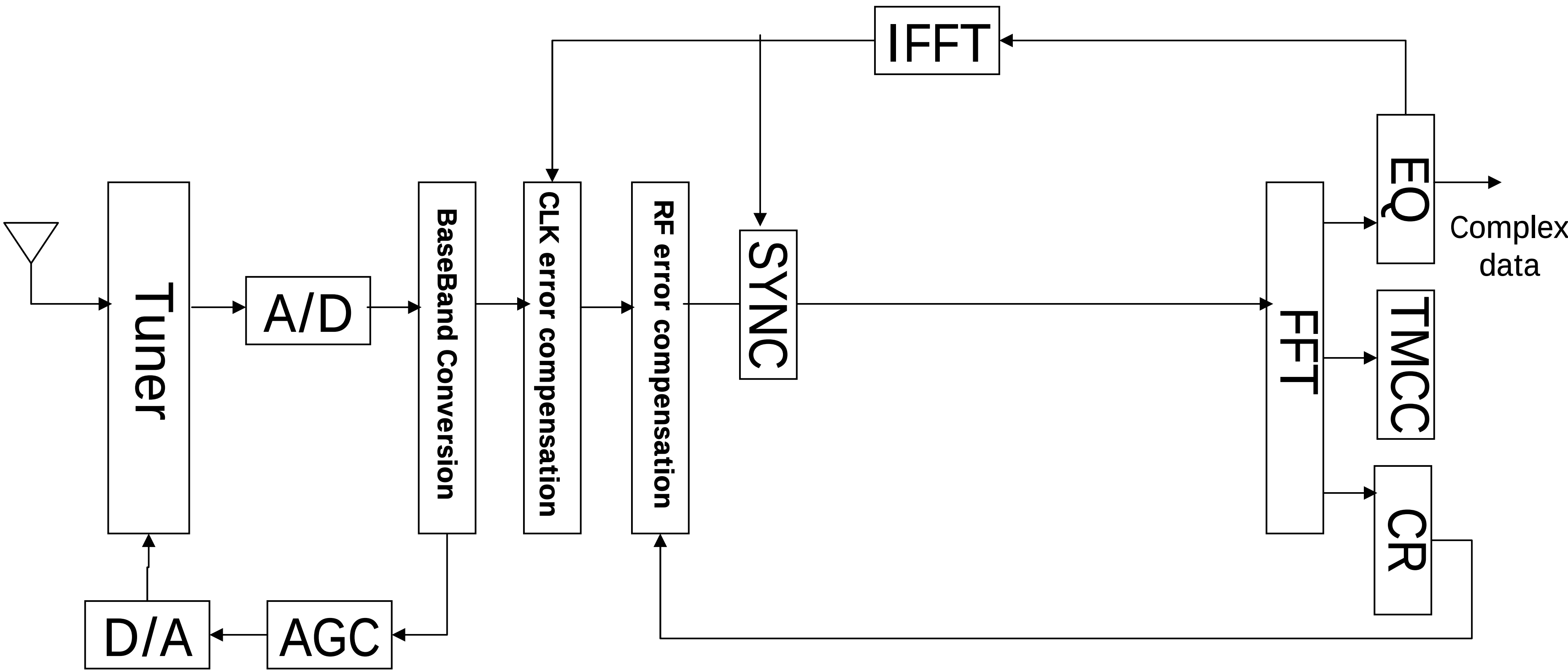
- 225MHz (1350MFLOPS)
- Internal Memory
  - **Program Area：4KB**
  - **Data Area：4KB**
  - **SRAM：192KB**

Peripheral

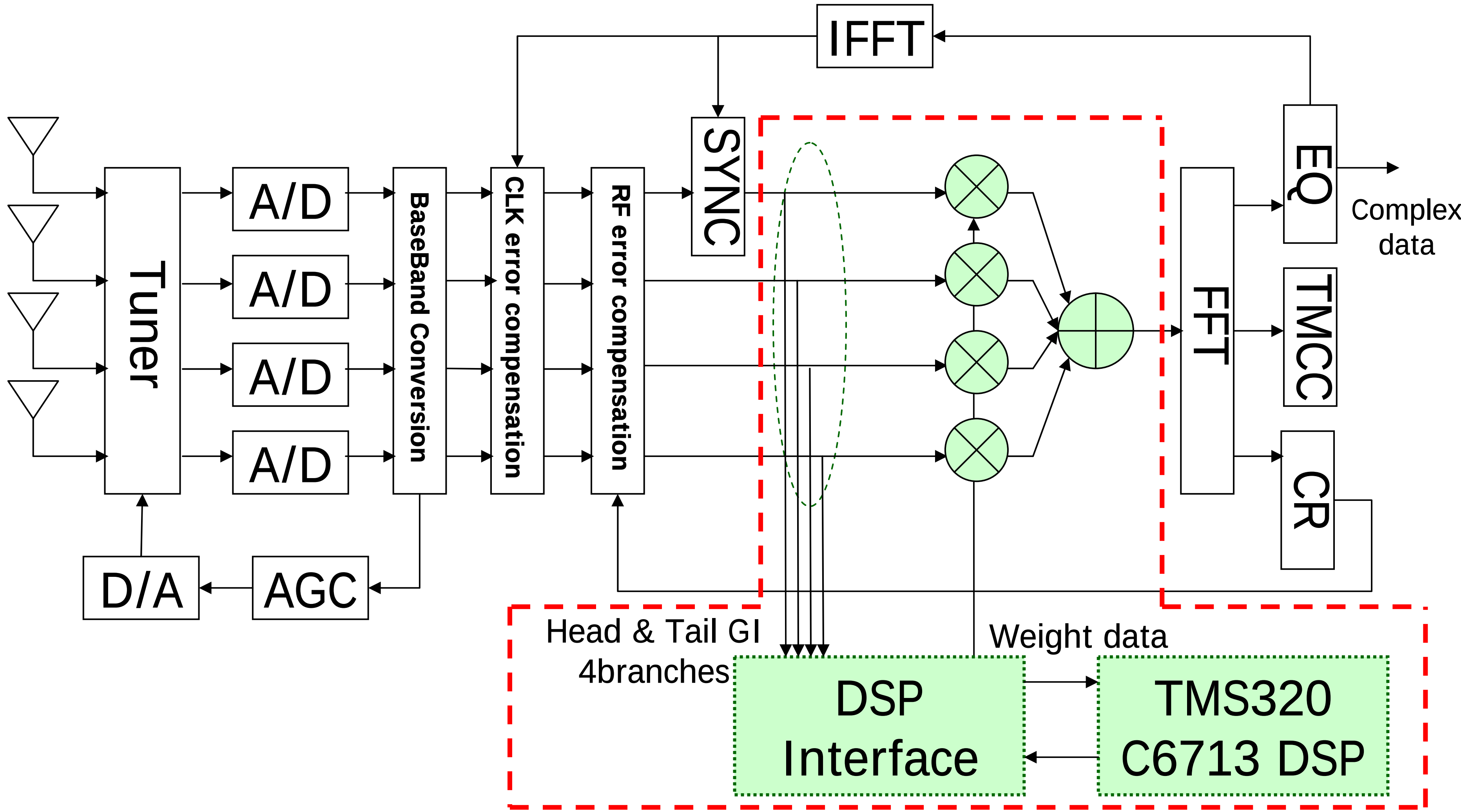
- 32bit EMIF
- GPIO



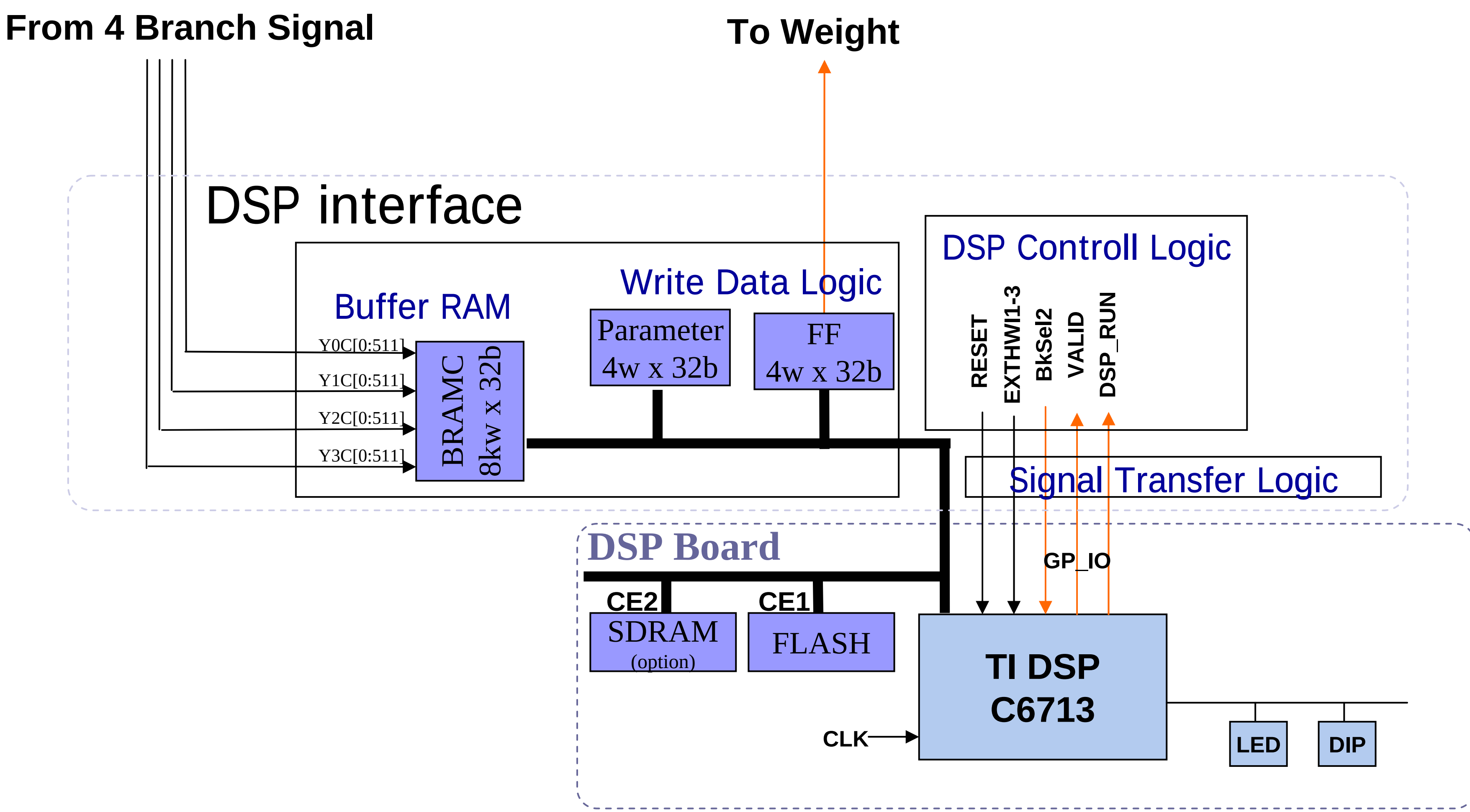
1 antenna OFDM Receiver



4 antenna DSP based AAA OFDM receiver



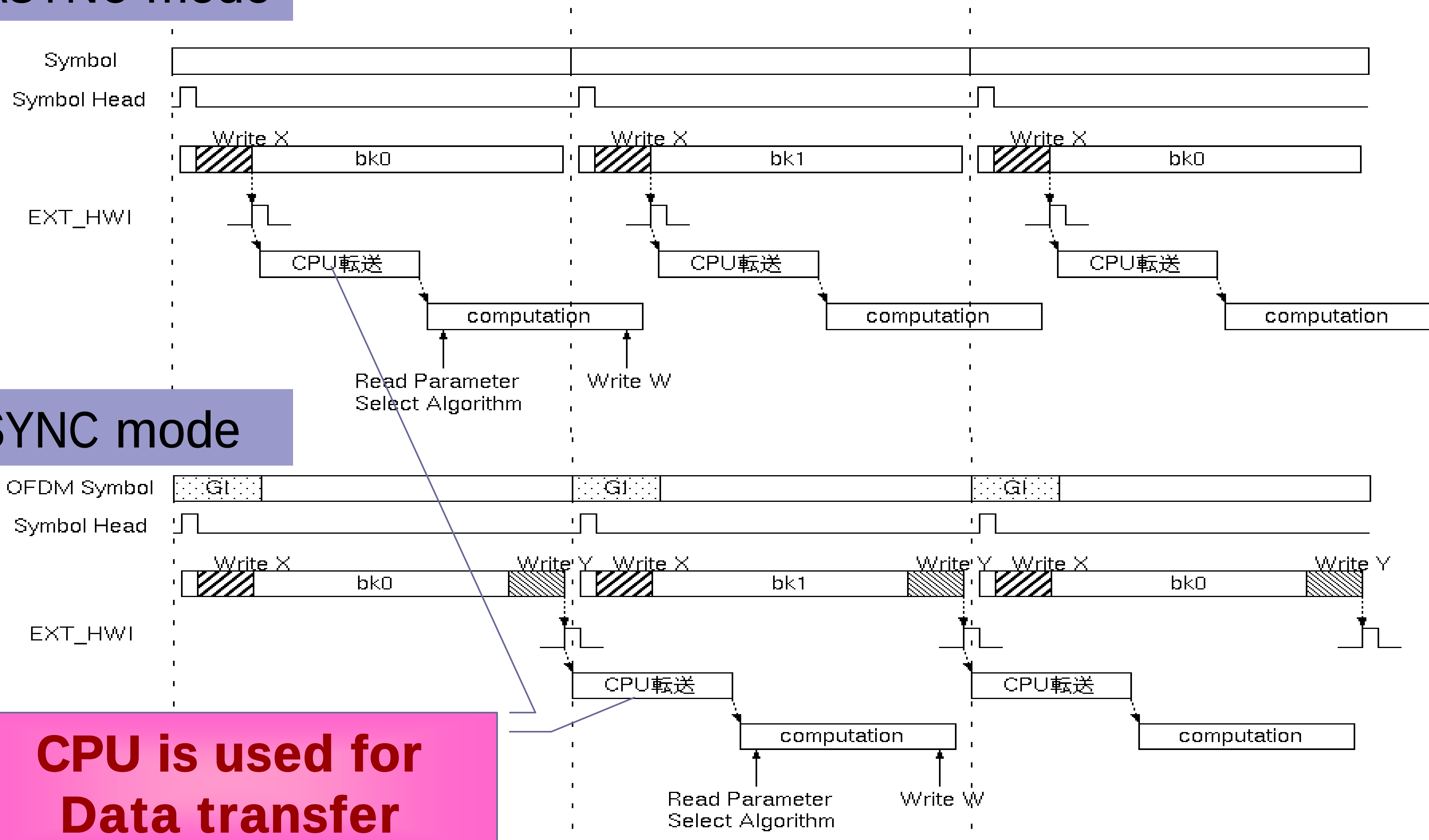
DSP Interface



H/W – S/W interface timing diagram w/o DMA

ASYNC mode

SYNC mode



CPU is used for Data transfer

Performance Optimization

- Let processor core to concentrate weight calculation!

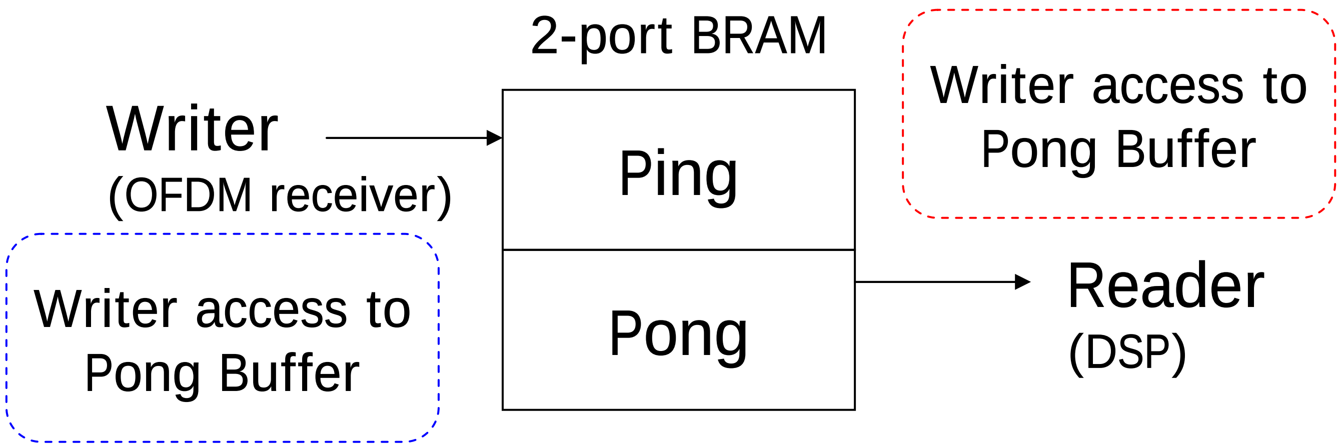
⬇

  - EDMA (Enhanced Direct Memory Access)
    ➡ CPU core is free for Data transfer

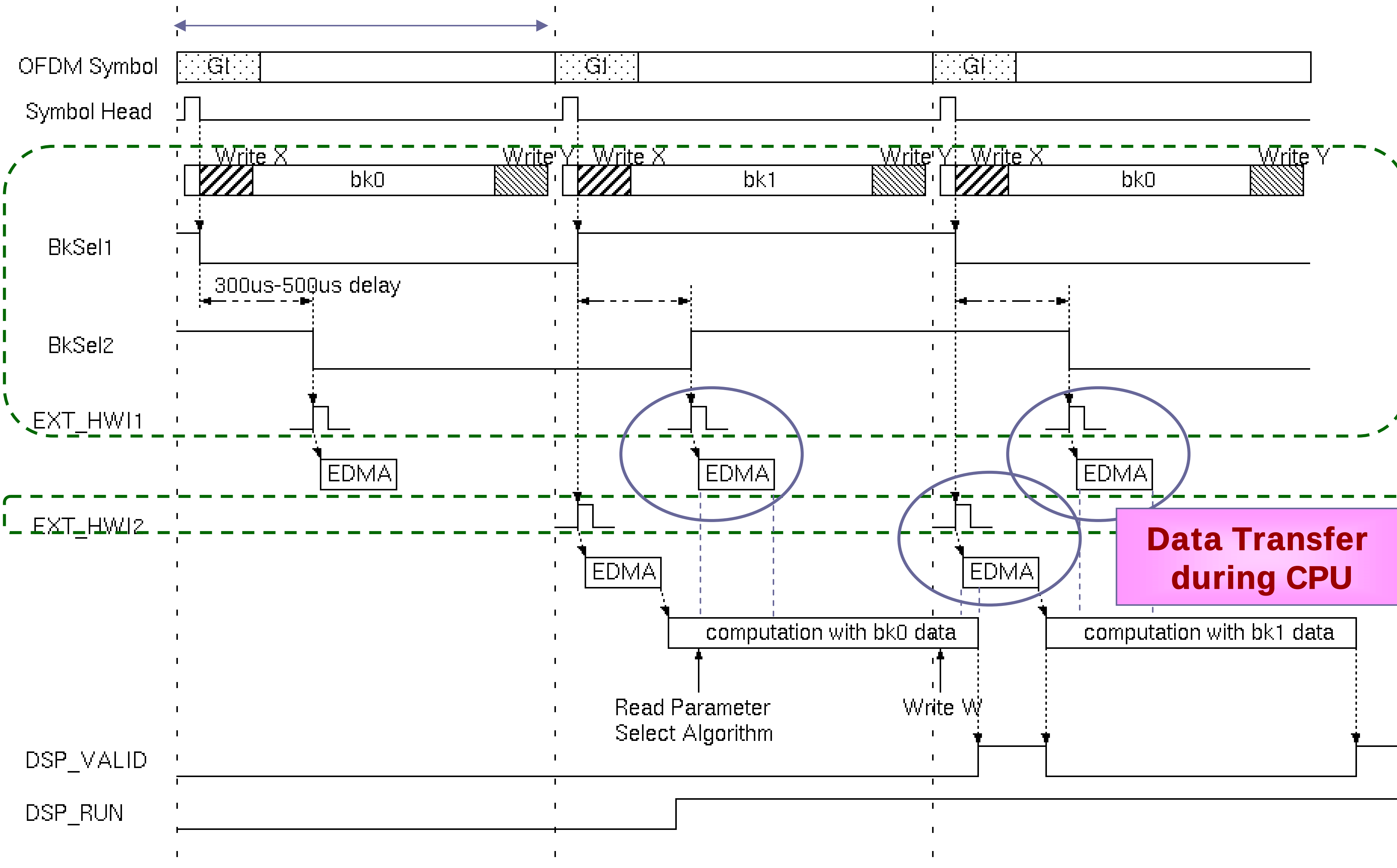
  - Double memory buffer in DSP
    ➡ EDMA memory access does NOT conflict with CPU core memory access.

Double Buffer

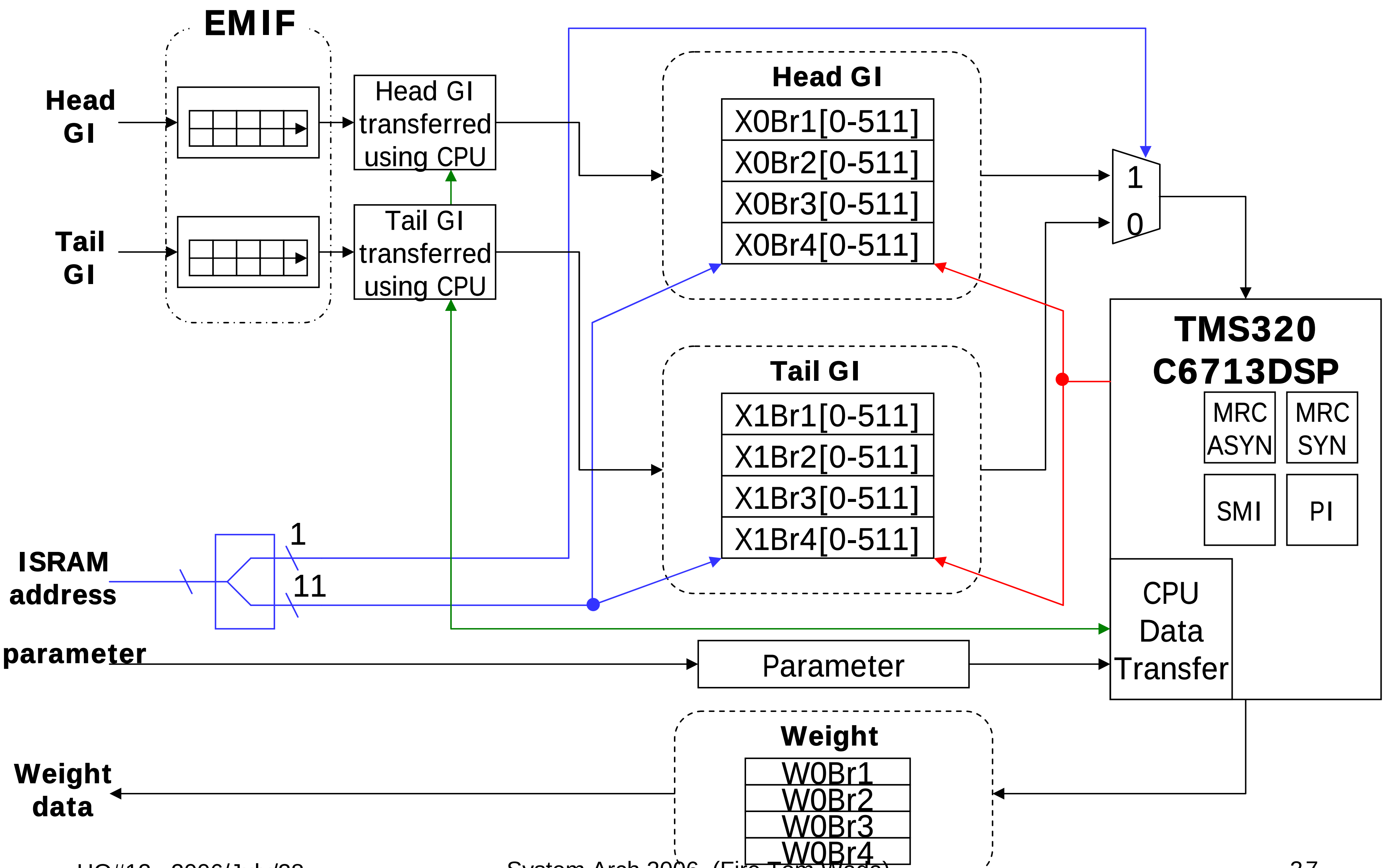
- 2 bank Ping-Pong buffer
- 2-port RAM is used for Real Implementation.
  - Each Port can operate at Different CLK frequency.



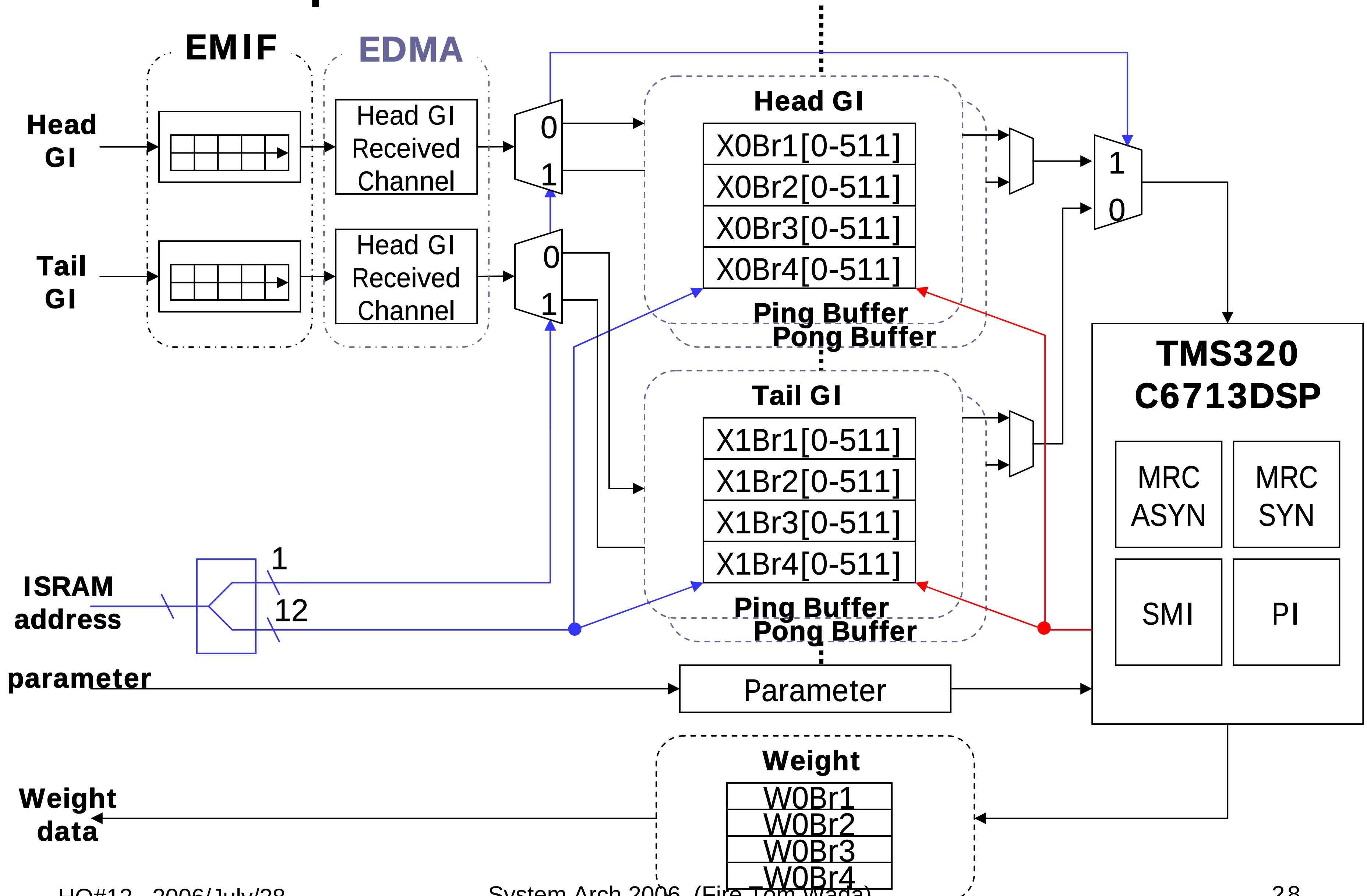
H/W – S/W interface timing diagram



Before Optimization



After Optimization



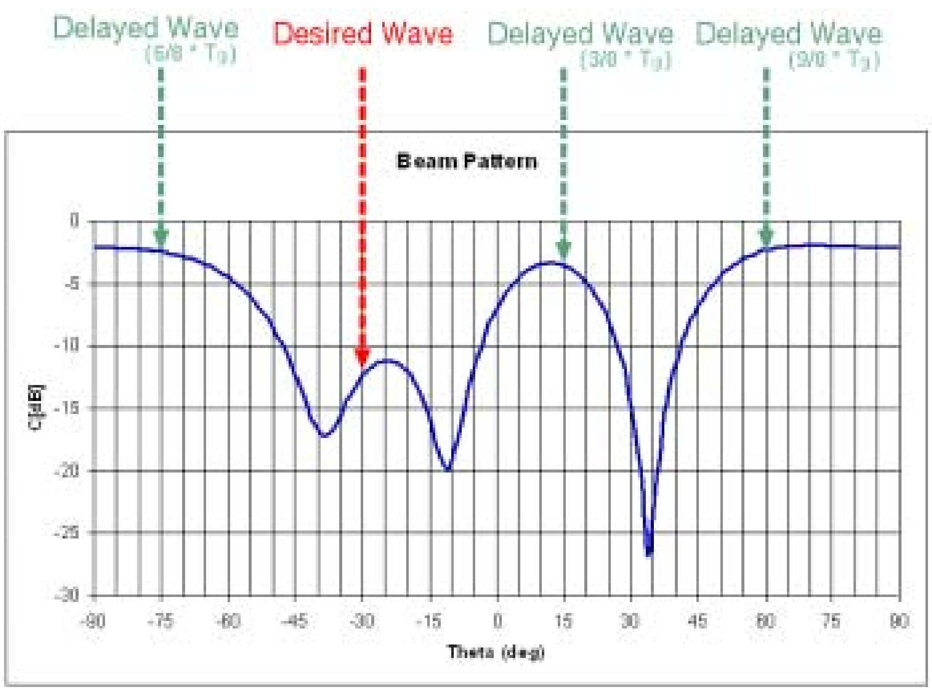
CPU Speed Comparison

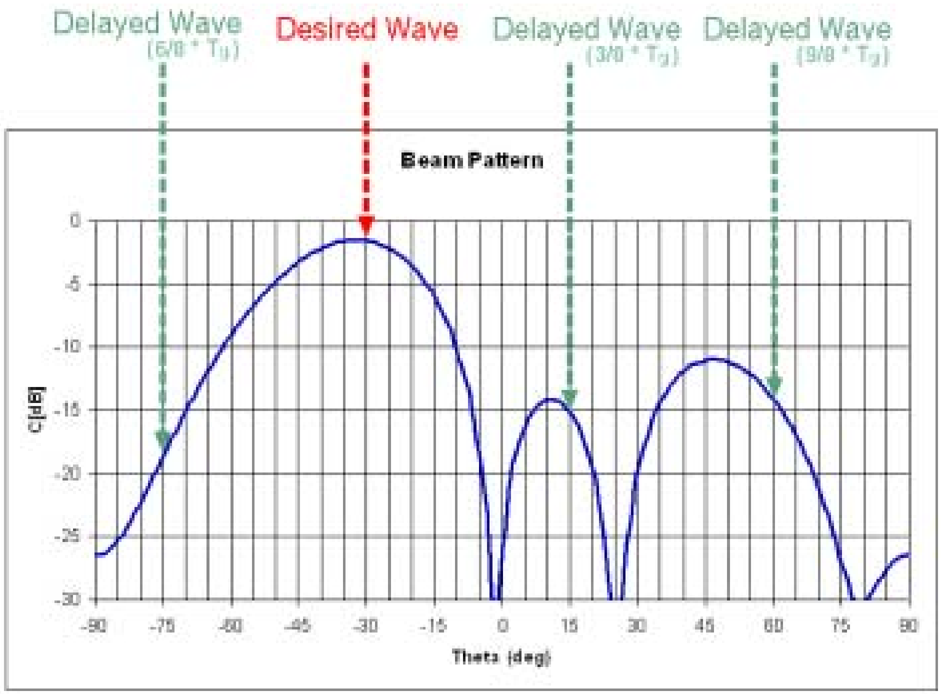
	ASYNC mode	SYNC mode		
	MRC_ASYN	MRC_SYN	SMI	PI
Before	343.58μs	364.99μs	470.19μs	413.98μs
After	147.54μs	173.64μs	268.15μs	223.95μs
Improvement	57.06%	52.43%	42.97%	45.90%

1 EDMA per Symbol (MRC_ASYN) **2 EDMA per Symbol** (MRC_SYN, SMI, PI)

MAX 57% Speed Enhancement

Measured Results [MRC_ASYN, MRC_SYN)]

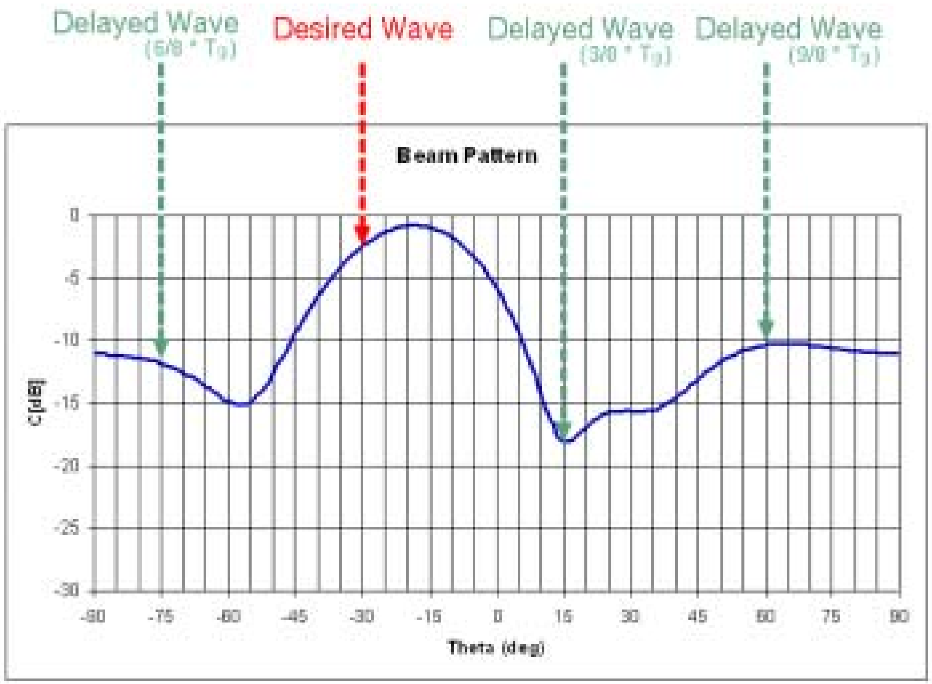




MRC_ASYN方式
BER: 1.30E-02

MRC_SYN方式
BER: 4.3E-03

Measured Results [SMI,PI]



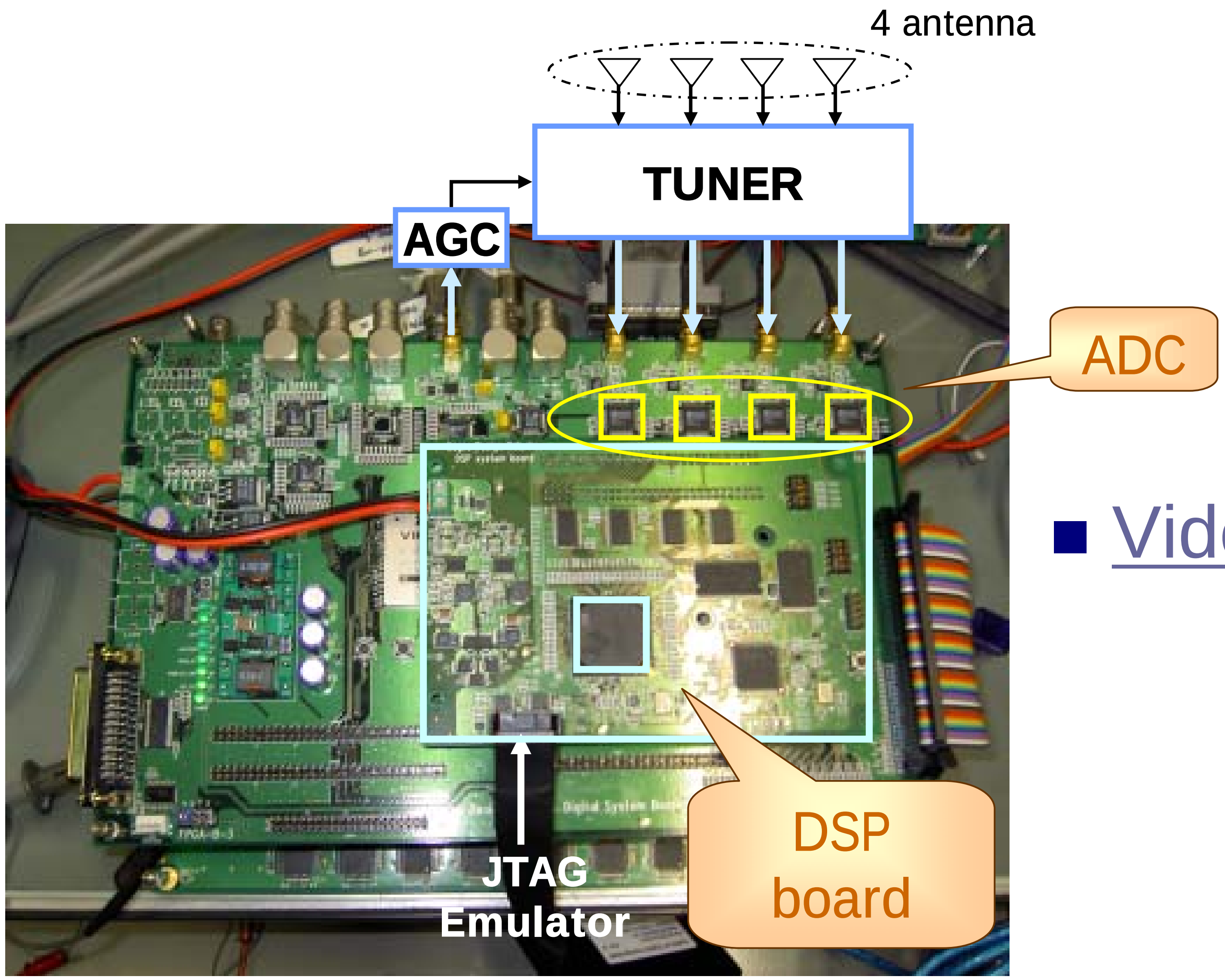
SMI方式

BER: 6.60E-03

PI方式

BER: 2.40E-03

SYSTEM PHOTOGRAPH



- Video

- ALL SUBJECTS ARE FINISHED!

- THANK YOU!!!