

# Computer-Aided VLSI System Design

## Final Project:

# 5G MIMO Demodulation: QR Decomposition

Lecturer: Yi-Lin Lo

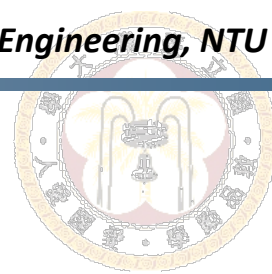
*Graduate Institute of Electronics Engineering, National Taiwan University*

MediaTek



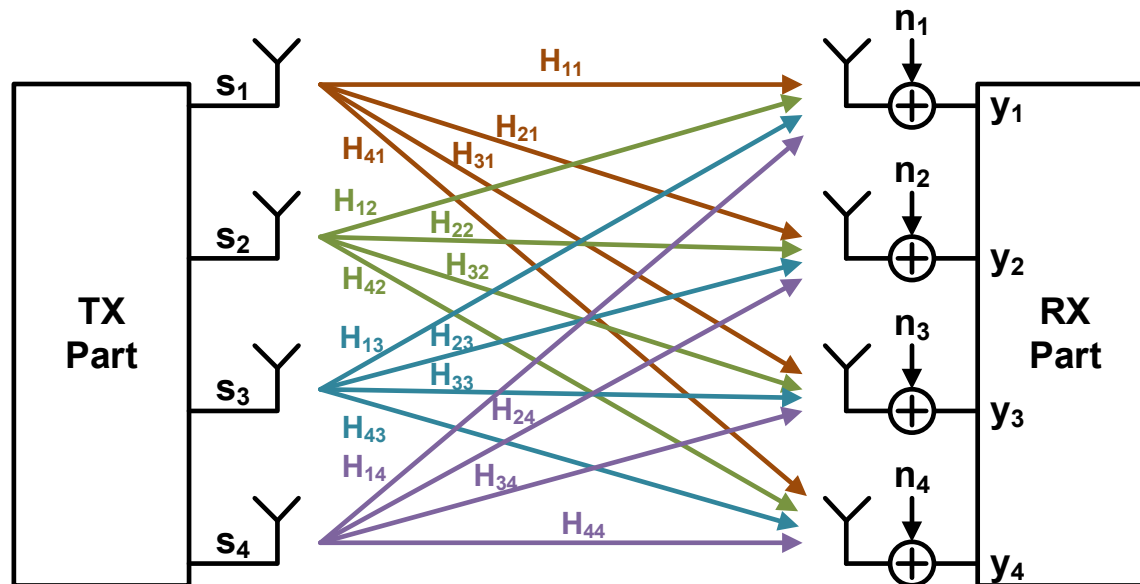
NTU GIEE



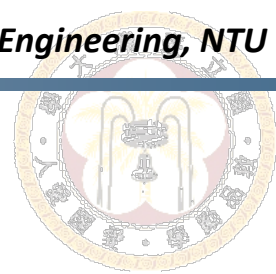


# Overview

- **MIMO** (multiple input, multiple output) is an antenna technology for wireless communications, the encode flow is as follows:



- At the receiver, we need to decode the data by reverting the encode flow
- In this project, we'll try to implement a part of simple **MIMO receiver** to demodulate the RX data
  - AWGN (additive white Gaussian noise) channel



# System Model

- The received signal  $\underline{y}$  per data RE can be expressed as [1] **RE: resource element**

- $\underline{y} = \underline{H}\tilde{\underline{s}} + \underline{n}$

- $\underline{H}$ : channel,  $\tilde{\underline{s}}$ : transmitted symbol,  $\underline{n}$ : noise

- At the 4TX \* 4RX transmission, the formula can be re-written as

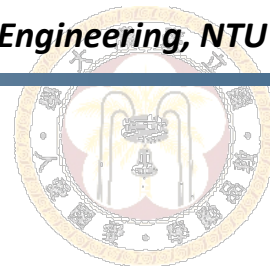
- $$\begin{bmatrix} \underline{y}_1 \\ \underline{y}_2 \\ \underline{y}_3 \\ \underline{y}_4 \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} & H_{14} \\ H_{21} & H_{22} & H_{23} & H_{24} \\ H_{31} & H_{32} & H_{33} & H_{34} \\ H_{41} & H_{42} & H_{43} & H_{44} \end{bmatrix} \begin{bmatrix} \tilde{\underline{s}}_1 \\ \tilde{\underline{s}}_2 \\ \tilde{\underline{s}}_3 \\ \tilde{\underline{s}}_4 \end{bmatrix} + \begin{bmatrix} \underline{n}_1 \\ \underline{n}_2 \\ \underline{n}_3 \\ \underline{n}_4 \end{bmatrix}$$

- The MIMO receiver is to demodulate the  $\tilde{\underline{s}}$  by  $\underline{y}$ ,  $\underline{H}$ ,  $\underline{n}$ .

- $\tilde{\underline{s}}_1, \tilde{\underline{s}}_2, \tilde{\underline{s}}_3$  and  $\tilde{\underline{s}}_4$  are the symbol with modulation (**QPSK – 2-bit data**)

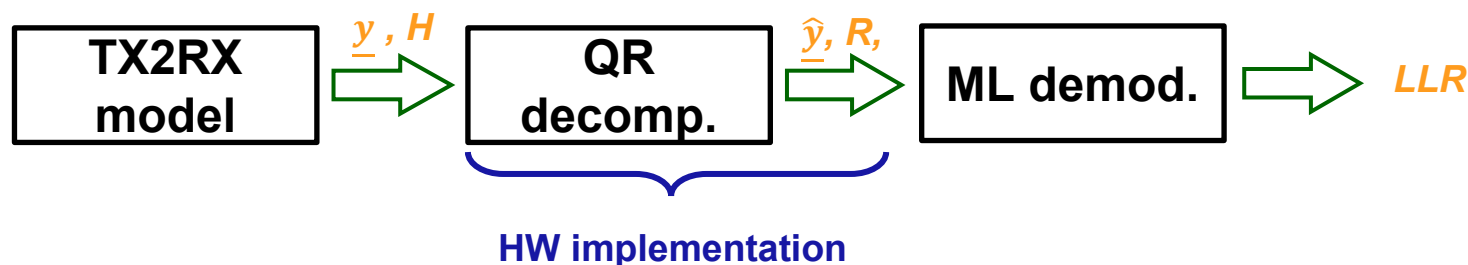
- The output of MIMO receiver is the LLR per bit

- LLR : log likelihood ratio, if the value is positive, it means the possibility of this bit is 0 is much higher than 1, and vice versa
    - Total **8 LLRs** per data RE (4-signal \* 2-bit (QPSK))

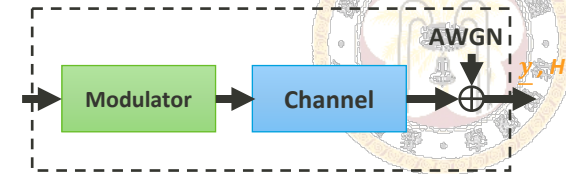


# System Model

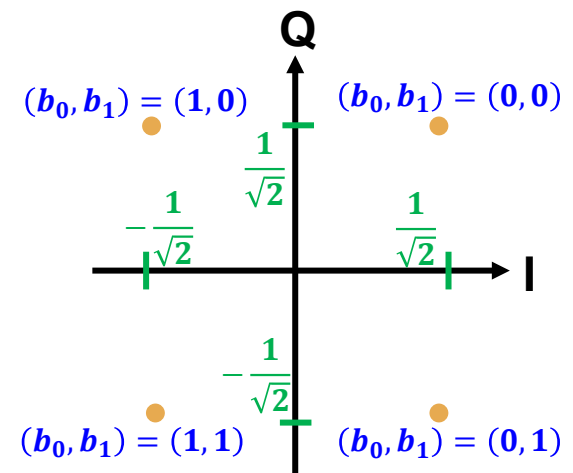
- From [1], MIMO receiver is composed of QR decomposition (QRD) and Maximum Likelihood (ML) demodulation

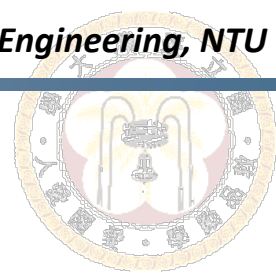


# Tx2Rx Model



- Tx2Rx Model
  - $\underline{y} = \underline{H}\tilde{\underline{s}} + \underline{n}$
- Modulator: transmitted signal  $\tilde{\underline{s}}$ 
  - QPSK: pairs of bits are mapped to complex-valued modulation symbols
- MIMO
  - Channel: multiply by channel matrix:  $\underline{H}$ 
    - 4X4 matrix, complex number
    - Normal distribution random matrix
    - $\underline{H} \sim N(0, 1/4)$
  - AWGN: add noise  $\underline{n}$ 
    - adds white Gaussian noise
    - $\underline{n} \sim N(0, 1)$





# QR Decomposition (QRD)

## ■ Motivation

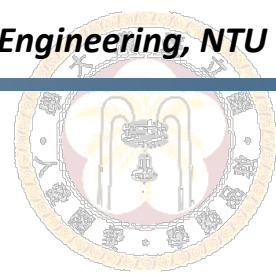
- Reduce the complexity of Maximum Likelihood (ML) demodulation

- $\hat{\underline{s}} = \underset{\underline{s} \in A}{\operatorname{argmin}} (\|\underline{y} - H\underline{s}\|^2)$  A: a set of all combinations of 4 transmitted symbol vectors ( $s_1 \sim s_4$ )

- With QR decomposition, a signal model can be re-written as

$$\begin{aligned}
 \underline{y} &= H\underline{s} + \underline{n} \\
 \underline{y} &= (\underline{Q}\underline{R})\underline{s} + \underline{n} \\
 \underline{Q}^H \underline{y} &= \underline{Q}^H \underline{Q} \underline{R} \underline{s} + \underline{Q}^H \underline{n} \\
 \hat{\underline{y}} &= \underline{R} \underline{s} + \underline{v}
 \end{aligned}
 \quad
 \begin{cases}
 \underline{Q}: \text{an } \textcolor{red}{orthogonal} \text{ matrix, where } \underline{Q}^H \underline{Q} = I \\
 \underline{R}: \text{an } \textcolor{green}{upper triangular} \text{ matrix}
 \end{cases}$$

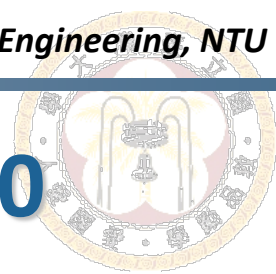
- ML demodulation question becomes  $\hat{\underline{s}} = \underset{\underline{s} \in A}{\operatorname{argmin}} (\|\hat{\underline{y}} - \underline{R}\underline{s}\|^2)$



# Modified Gram-Schmidt

- You can find the concept of Gram-Schmidt Procedure in Prof. Hung-Yi Lee's Linear Algebra course [2]
  - Starting from 21:30
- In this project, we use Modified Gram-Schmidt Procedure [3]
  - Please refer to **Numerical stability** in [3]

$$H = [h_1|h_2|h_3|h_4] = [e_1|e_2|e_3|e_4] \begin{bmatrix} h_1 \cdot e_1 & h_2 \cdot e_1 & h_3 \cdot e_1 & h_4 \cdot e_1 \\ 0 & h_2 \cdot e_2 & h_3 \cdot e_2 & h_4 \cdot e_2 \\ 0 & 0 & h_3 \cdot e_3 & h_4 \cdot e_3 \\ 0 & 0 & 0 & h_4 \cdot e_4 \end{bmatrix} = QR$$



# Modified Gram-Schmidt: Iteration 0

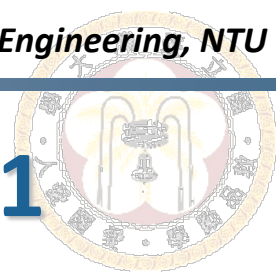
- 1. Calculate Euclidean distance  $\|h_1^{(0)}\|$  //  $R_{11}$
- 2. Calculate normalized orthogonal vector  $e_1 = h_1^{(0)} / \|h_1^{(0)}\|$
- 3. Calculate inner products
  - $h_2^{(0)} \cdot e_1 = e_1^H h_2^{(0)}$  //  $R_{12}$
  - $h_3^{(0)} \cdot e_1 = e_1^H h_3^{(0)}$  //  $R_{13}$
  - $h_4^{(0)} \cdot e_1 = e_1^H h_4^{(0)}$  //  $R_{14}$
- 4. Calculate orthogonal vectors  $h_2^{(1)}, h_3^{(1)}, h_4^{(1)}$  by removing its projection along  $h_1$ 
  - $h_2^{(1)} = h_2^{(0)} - proj_{h_1} h_2^{(0)} = h_2^{(0)} - (h_2^{(0)} \cdot e_1)e_1 = h_2^{(0)} - (R_{12})e_1$
  - $h_3^{(1)} = h_3^{(0)} - proj_{h_1} h_3^{(0)} = h_3^{(0)} - (h_3^{(0)} \cdot e_1)e_1 = h_3^{(0)} - (R_{13})e_1$
  - $h_4^{(1)} = h_4^{(0)} - proj_{h_1} h_4^{(0)} = h_4^{(0)} - (h_4^{(0)} \cdot e_1)e_1 = h_4^{(0)} - (R_{14})e_1$

$Q_{11}, Q_{21}, Q_{31}, Q_{41}$

Note: Euclidean distance  
 $= \sqrt{a^2 + b^2}$  for  $(a + bj)$

$$\text{proj}_{\mathbf{u}}(\mathbf{v}) = \frac{\langle \mathbf{v}, \mathbf{u} \rangle}{\langle \mathbf{u}, \mathbf{u} \rangle} \mathbf{u},$$

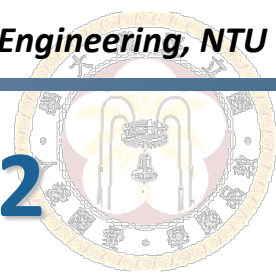




# Modified Gram-Schmidt: Iteration 1

- 1. Calculate Euclidean distance  $\|h_2^{(1)}\| // R_{22}$
- 2. Calculate normalized orthogonal vector  $e_2 = h_2^{(1)} / \|h_2^{(1)}\|$
- 3. Calculate inner products
  - $h_3^{(1)} \cdot e_2 = e_2^H h_3^{(1)} // R_{23}$
  - $h_4^{(1)} \cdot e_2 = e_2^H h_4^{(1)} // R_{24}$
- 4. Calculate orthogonal vectors  $h_3^{(2)}, h_4^{(2)}$  by removing its projection along  $h_2$ 
  - $h_3^{(2)} = h_3^{(1)} - proj_{h_2}^{h_3^{(1)}} = h_3^{(1)} - (h_3^{(1)} \cdot e_2)e_2 = h_3^{(1)} - (R_{23})e_2$
  - $h_4^{(2)} = h_4^{(1)} - proj_{h_2}^{h_4^{(1)}} = h_4^{(1)} - (h_4^{(1)} \cdot e_2)e_2 = h_4^{(1)} - (R_{24})e_2$

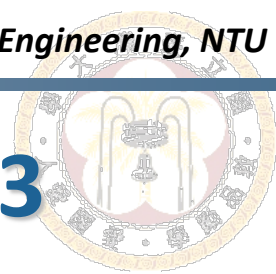
$Q_{12}, Q_{22}, Q_{32}, Q_{42}$



## Modified Gram-Schmidt: Iteration 2


- 1. Calculate Euclidean distance  $\|h_3^{(2)}\| // R_{33}$
- 2. Calculate normalized orthogonal vector  $e_3 = h_3^{(2)} / \|h_3^{(2)}\|$
- 3. Calculate inner products
  - $h_4^{(2)} \cdot e_3 = e_3^H h_4^{(2)} // R_{34}$
- 4. Calculate orthogonal vectors  $h_4^{(3)}$  by removing its projection along  $h_3$ 
  - $h_4^{(3)} = h_4^{(2)} - proj_{h_3}^{h_4^{(2)}} = h_4^{(2)} - (h_4^{(2)} \cdot e_3)e_3 = h_4^{(2)} - (R_{34})e_3$

↘  $Q_{13}, Q_{23}, Q_{33}, Q_{43}$



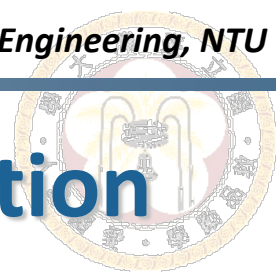
# Modified Gram-Schmidt: Iteration 3

- 1. Calculate Euclidean distance  $\|h_4^{(3)}\| // R_{44}$
- 2. Calculate normalized orthogonal vector  $e_4 = h_4^{(3)} / \|h_4^{(3)}\|$


 $Q_{14}, Q_{24}, Q_{34}, Q_{44}$

- Received signal  $\underline{\hat{y}}$  ( $\underline{\hat{y}}$ ) becomes

$$\underline{\hat{y}} = Q^H \underline{y} = \begin{bmatrix} Q_{11}^* & Q_{21}^* & Q_{31}^* & Q_{41}^* \\ Q_{12}^* & Q_{22}^* & Q_{32}^* & Q_{42}^* \\ Q_{13}^* & Q_{23}^* & Q_{33}^* & Q_{43}^* \\ Q_{14}^* & Q_{24}^* & Q_{34}^* & Q_{44}^* \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} Q_{11}^* y_1 + Q_{21}^* y_2 + Q_{31}^* y_3 + Q_{41}^* y_4 \\ Q_{12}^* y_1 + Q_{22}^* y_2 + Q_{32}^* y_3 + Q_{42}^* y_4 \\ Q_{13}^* y_1 + Q_{23}^* y_2 + Q_{33}^* y_3 + Q_{43}^* y_4 \\ Q_{14}^* y_1 + Q_{24}^* y_2 + Q_{34}^* y_3 + Q_{44}^* y_4 \end{bmatrix}$$



# Maximum Likelihood (ML) Demodulation

## ■ Soft-bit calculation [1]

$$LLR: L(x_{k,b} | \underline{y}) \approx \min_{\underline{x} \in X_{k,b,1}} \|\underline{y} - H\underline{s}\|^2 - \min_{\underline{x} \in X_{k,b,0}} \|\underline{y} - H\underline{s}\|^2 \quad \underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} (x_{1,1}, x_{1,2}) \\ (x_{2,1}, x_{2,2}) \\ (x_{3,1}, x_{3,2}) \\ (x_{4,1}, x_{4,2}) \end{bmatrix} \Leftrightarrow \underline{s}$$

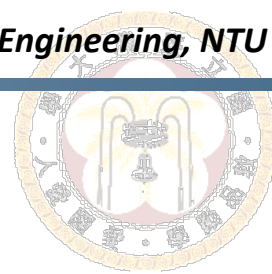
$\downarrow$   
k<sup>th</sup> entry of  $\underline{x}$ ,  
b<sup>th</sup> bit in the  $x_k$ 
 $\downarrow$   
 $X_{k,b,1}$ : subsets of  $\{\underline{x}\}$  with the b<sup>th</sup> bit in the k<sup>th</sup> entry = 1  
 $X_{k,b,0}$ : subsets of  $\{\underline{x}\}$  with the b<sup>th</sup> bit in the k<sup>th</sup> entry = 0

$$\|\underline{y} - H\underline{s}\|^2 \Rightarrow \|\hat{\underline{y}} - R\underline{s}\|^2 = \left( \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \\ \hat{y}_4 \end{bmatrix} - \begin{bmatrix} R_{11} & R_{12} & R_{13} & R_{14} \\ 0 & R_{22} & R_{23} & R_{24} \\ 0 & 0 & R_{33} & R_{34} \\ 0 & 0 & 0 & R_{44} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} \right)^2$$

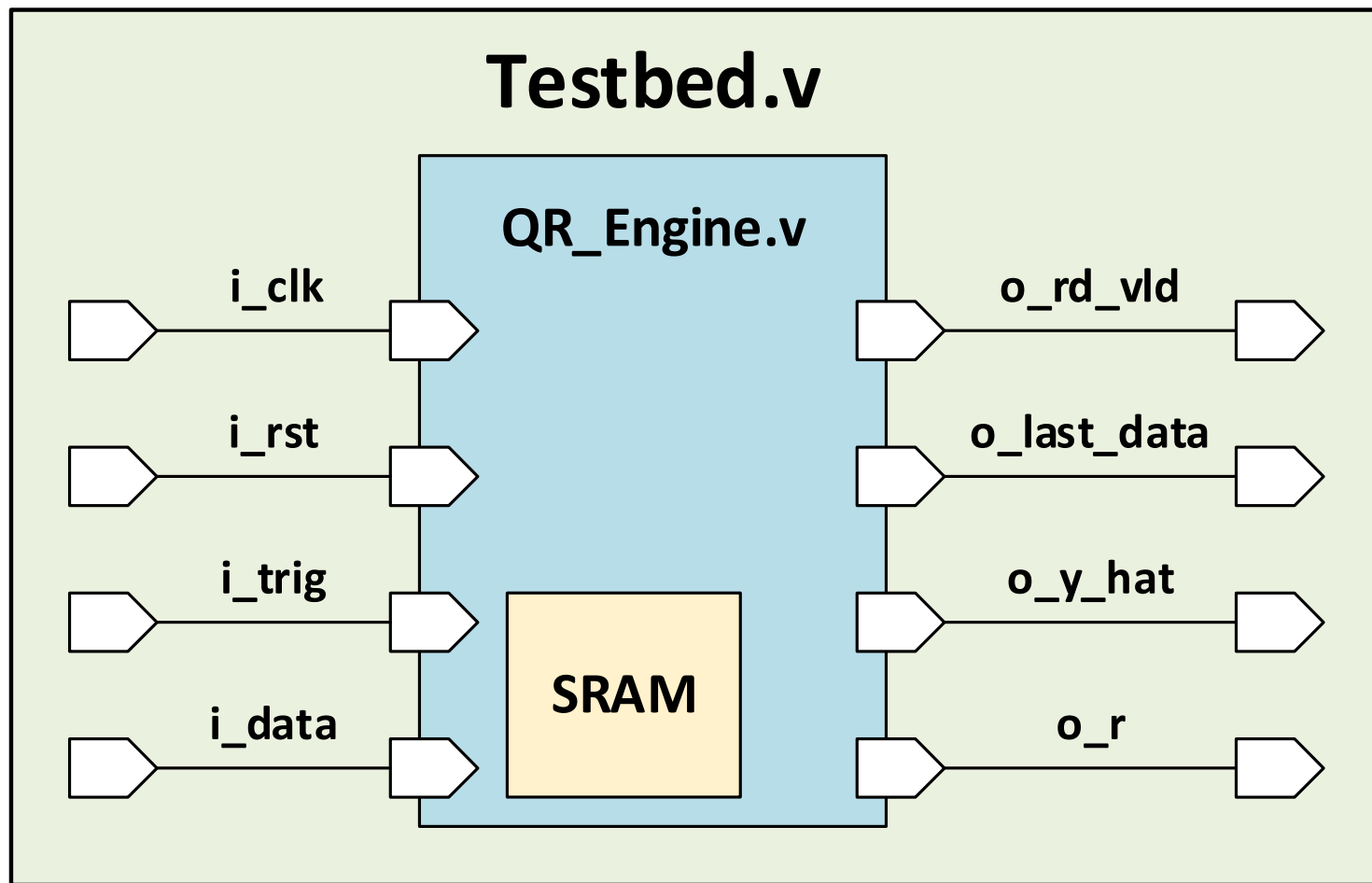
$$= \sum_{i=1}^4 \left| \hat{y}_i - \sum_{j=1}^4 R_{ij} s_j \right|^2$$

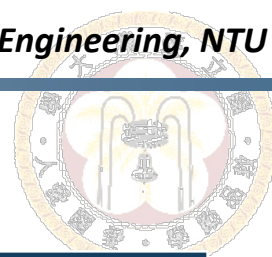
## ■ Hard-bit calculation

- $L(x_{k,b} | \underline{y})$  's sign-bit = 0, hard-bit out = 0
- $L(x_{k,b} | \underline{y})$  's sign-bit = 1, hard-bit out = 1



# Block Diagram

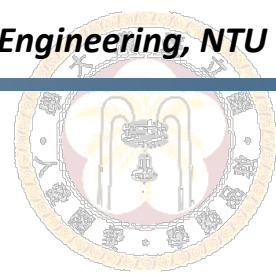




# Input/Output

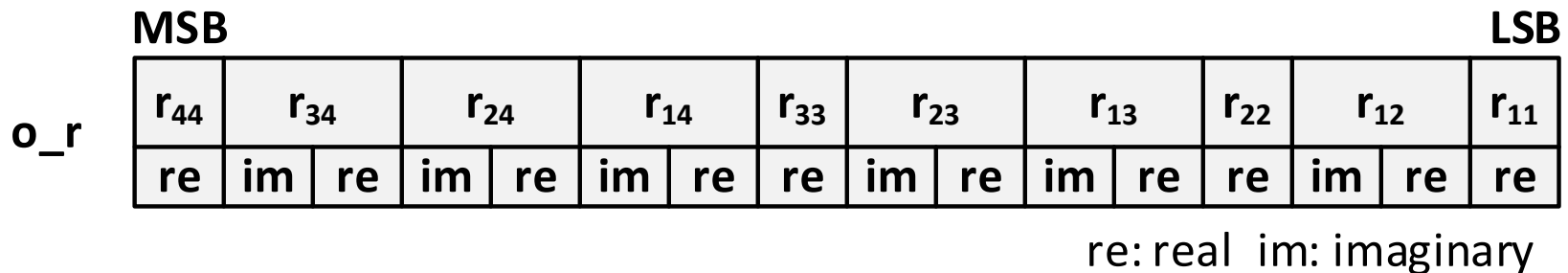
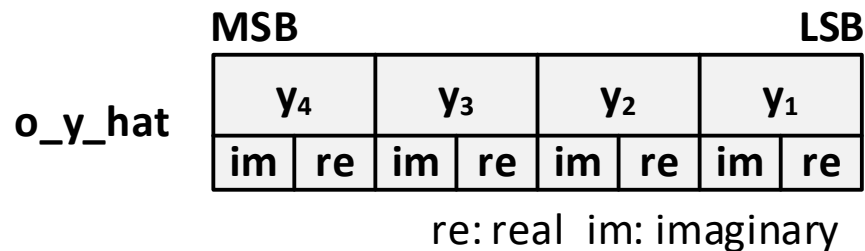
Signal Name	I/O	Width	Simple Description
i_clk	I	1	本系統為同步於時脈正緣之同步設計。 (註: Host端採clk正緣時送資料。)
i_rst	I	1	高位準 “非” 同步(active high asynchronous)之系統重置信號。
i_trig	I	1	輸入資料有效控制訊號。當為high時i_data有效。
i_data	I	48	每筆資料包含虛部與實部({imaginary, real})，各24位元，為{S1.22}之fixed point。
o_rd_vld	O	1	輸出資料有效之控制訊號。當為High時，表示目前輸出的 o_y_hat 與 o_r 為有效的。
o_last_data	O	1	當為High時，下個i_clk正緣i_trig為High開始送資料(200 cycles)
o_y_hat	O	160	y_hat資料輸出，包含 4 筆，每筆各 40 bits，i_y_hat [159:120] 為 $y_4$ ，i_y_hat [119:80] 為 $y_3$ ，依此類推，每筆資料包含虛部與實部({imaginary, real})，各20位元，為{S3.16}之fixed point。
o_r	O	320	R資料輸出，依序為 $\{r_{44}, r_{34}, r_{24}, r_{14}, r_{33}, r_{23}, r_{13}, r_{22}, r_{12}, r_{11}\}$ ， $r_{ij}$ 僅包含實部，為20位元，{S3.16}之fixed point， $r_{ij}$ 則包含虛部與實部({imaginary, real})，各20位元，同樣為{S3.16}之fixed point。

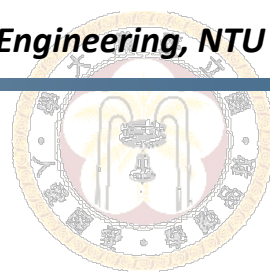
{SA.B}: fixed point with sign bit, A-bit integer, and B-bit fraction



## Data format

- o\_y\_hat and o\_r
  - Real and imaginary are both S3.16

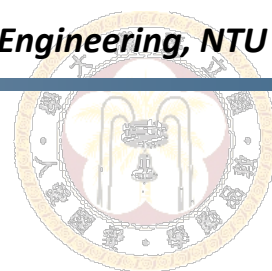




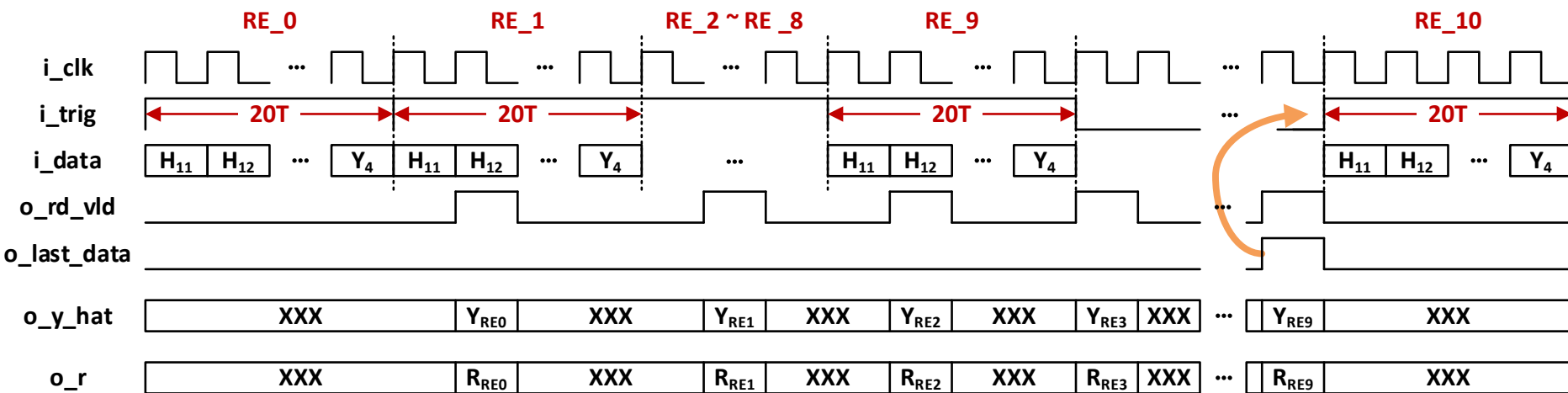
## I/O Spec

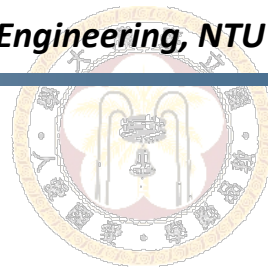
- Input:
  - i\_trig: comes in consecutive 200 cycles, each cycle carries one element of H and y
  - i\_data: complex number, s1.22 for each element
    - Order:  $H_{11} \rightarrow H_{12} \rightarrow H_{13} \rightarrow H_{14} \rightarrow Y_1 \rightarrow H_{21} \rightarrow H_{22} \rightarrow H_{23} \rightarrow H_{24} \rightarrow Y_2$   
 $\rightarrow H_{31} \rightarrow H_{32} \rightarrow H_{33} \rightarrow H_{34} \rightarrow Y_3 \rightarrow H_{41} \rightarrow H_{42} \rightarrow H_{43} \rightarrow H_{44} \rightarrow Y_4$
- Output:
  - o\_rd\_vld: set high (a RE of R and y\_hat are ready)
  - o\_last\_data: set high (last RE(#10) of R and y\_hat are ready)
  - R: s3.16 for each element
  - y: s3.16 for each element





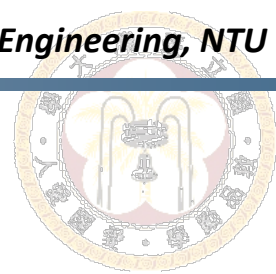
# Spec - Interface





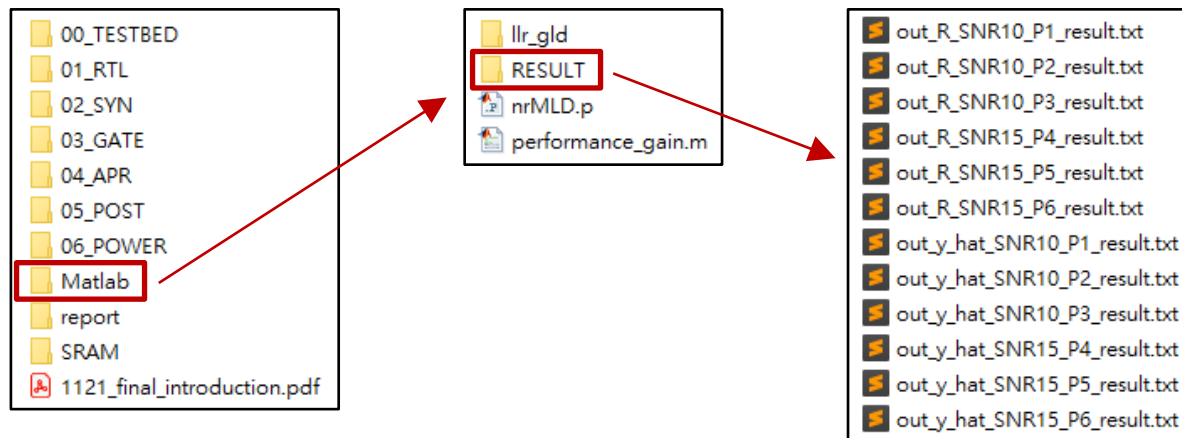
# Specification

- **The clock frequency should be at least 200 MHz**
- Only worst-case library is used for synthesis and APR.
- The slack for setup-time should be non-negative.
- **No any timing violation and glitches** for the gate level simulation and post-layout simulation.



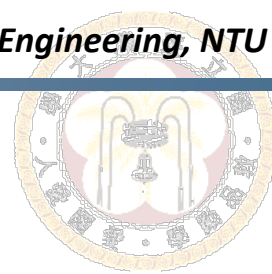
# Design Files

- Use the functions in the “Matlab” file to evaluate the soft LLR error rate of the files generated by RTL simulation



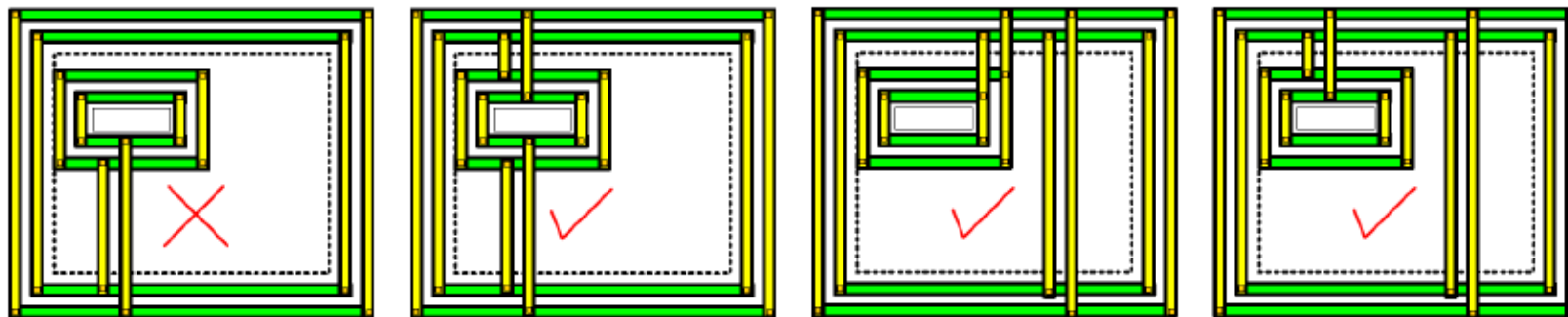
- Change the “packet\_no” to evaluate files from packets

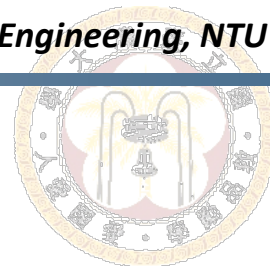
```
performance_gain.m  x  +
1  clear
2  close all
3  clc
4
5  % Change the packet number (1-6)
6  packet_no = 6;
7
8  switch packet_no
9      case 1
10         file_R = importdata('.\Result\out_R_SNR10_P1_result.txt');
11         file_y_hat = importdata('.\Result\out_y_hat_SNR10_P1_result.txt');
12         llr_gld = importdata('.\llr_gld\packet_1\llr_gld.mat');
```



# Specifications for APR (1)

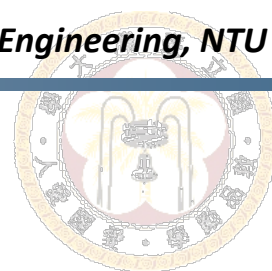
- 只需做 Marco layout 即不用包含 IO Pad 、 Bonding Pad)
- VDD 與 VSS Power Ring 寬度請各設定為 2um 只須做一組
- 不需加 Dummy Metal
- Power Stripe 務必至少加一組，其 VDD 、 VSS 寬度各設定為 2um
  - Power Stripe 垂直方向至少一組，水平方向可不加





## Specifications for APR (2)

- 務必要加 Power Rail (follow pin)
- Core Filler 務必要加
- APR 後之 GDSII 檔案務必產生
- 完成 APR DRC/LVS 完全無誤
- 記得先產生QR\_Engine.ioc，再重新讀取該檔來設定 pin position

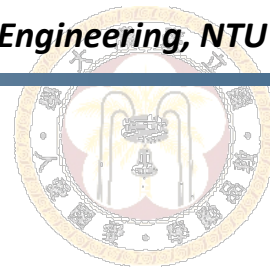


# Grading Policy

- **Baseline** 50% + **Performance** 40% + **Report** 10%

Item	%	Description
RTL Simulation	20	Pass full pattern simulation with specs
Synthesis	10	Pass gate-level sim
APR	20	Finish APR with no DRC/LVS errors Pass post-layout simulation
Performance	40	Area x Time x Power / (Performance Gain)
Report	10	1. Algorithm 2. Hardware implementation

Violation	Penalty
Clock Frequency of post-sim < 200 MHz	Performance*0.5
Gate-level sim pass but post-sim fail	Performance*0.5
Only RTL pass	Performance不評分
違反繳交格式與規則	總分-5



# Grading Policy - Test Pattern

- Total 6 packets: P#1-3 (SNR 10dB) and P#4-6 (SNR 15dB)
- **Another 6 packets of hidden data: P#7-9 and P#10-12**

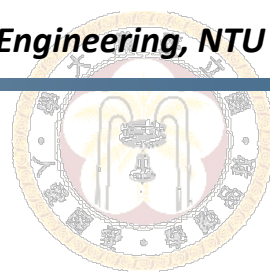
Soft LLR Error Rate (SNR)					
Soft LLR Error Rate (10 dB)	< 2 %	< 6%	< 10 %	< 13%	>= 13%
Soft LLR Error Rate (15 dB)	< 1%	< 2%	< 4%	< 6%	>= 6%
<b>Performance Gain</b>	1.76x	1.32x	1.14x	1.00x	fail

– For example, if a student gets

- 1.56% in P#1, 1.12% in P#2, 1.00% in P#3, 1.20% in P#7, 1.30% in P#8, **2.01%** in P#9 -> Performance Gain 1.32x
- 3.57% in P#4, **4.13%** in P#5, 1.56% in P#6, 3.50% in P#10, 2.57% in P#11, 0.56% in P#12 -> Performance Gain 1.00x
- Take average, Performance Gain  $(1.32+1.00)/2 = 1.16x$  as final score

- Total 1000 data RE at each packet

See soft LLR error rate in appendix



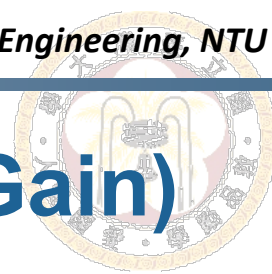
# Power

- Only need to calculate the total power of 10 RE
- Change the parameter “NO\_10RE” to **1** in testbench

```
testfixture.v
1  `timescale 1ns/1ps
2  `define CYCLE 5.0
3
4  // NO_10RE 100 (function check)
5  // NO_10RE 1   (power analysis)
6  `define NO_10RE 1 // 1 packet = 1000RE
7
8  module testfixture;
```

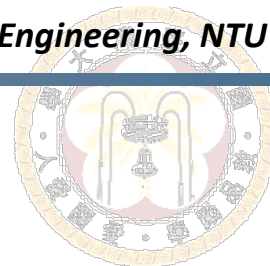
- Use the output fsdb file for power analysis of primetime





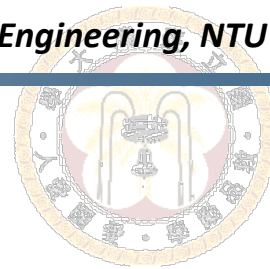
# Area x Time x Power / (Performance Gain)

- Area ( $\mu\text{m}^2$ ): Core area
  - 小數點以下四捨五入到兩位
- Time (ns): Simulation time of postsim
  - 整數
- Power (mW): Total power of postsim
  - 小數點以下四捨五入到一位



# Grading Policy - Report

- Algorithm
  - QR decomposition algorithm introduction
  - FXP setting
  
- HW implementation
  - HW scheduling
  - HW block diagram
  - Area / Power / Latency report
    - Technique sharing for HW improvement

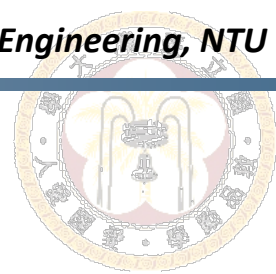


# Submission Files

- **Due Tuesday, Dec. 19, 23:59 (Submit to NTUCOOL)**
  - **No late submission**
- **Require data (with the required directory hierarchy):**

Violation	Penalty
01_RTL	1. All design Verilog files 2. rtl.f
02_SYN	1. Area/timing reports
03_GATE	1. QR_Engine_syn.v/sdf 2. rtl.f
04_APR	1. Route database 2. QR_Engine.gds
05_POST	1. QR_Engine_pr.v/sdf 2. rtl.f
reports	1. design.spec. 2. teamXX_report.pdf

- **Final project presentation (MTK experience sharing)**
  - **Date: Dec. 26, 2023**



# design.spec

Maximum operating frequency: (MHz)

Performance Gain:

POST-SIM cycle: (ns)

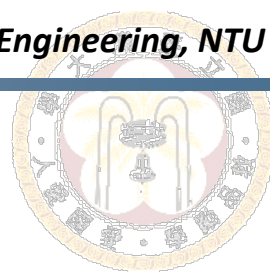
POST-SIM latency: (ns)

Post layout area: (um<sup>2</sup>)

Post layout total power: (mW)

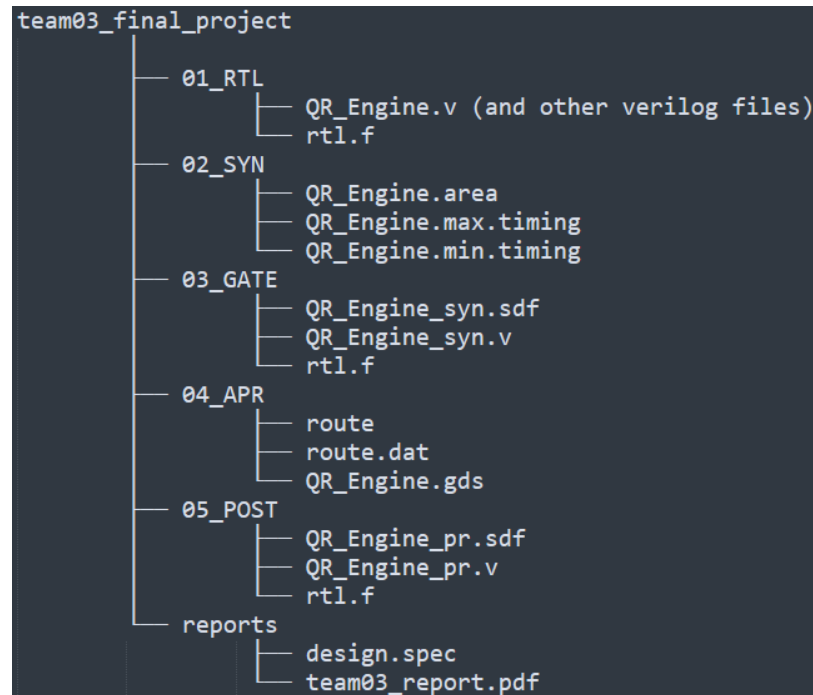
# of DRC violations:

Status of LVS check: (pass/fail)

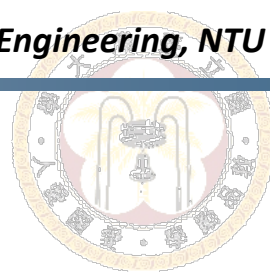


# Submission Hierarchy

- Folder name: teamID\_final\_project. Follow the hierarchy below

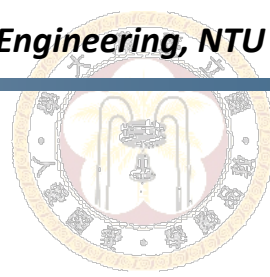


- Compress the folder teamID\_final\_project in a tar file named teamID\_final\_project\_vk.tar (k is the number of version, k =1,2,...), e.g. team03\_final\_project\_v1.tar



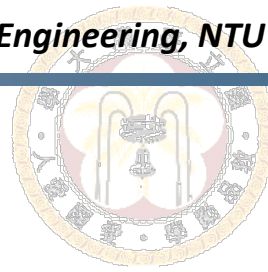
# Final Project Presentation

- The **top 5-6 groups** have the opportunity to give a presentation on stage and will be eligible for **additional bonus points**
  - PowerPoint
  - Approximate 15 minutes
  
- Presentation Content
  - Algorithm (if you use other methods)
  - Bit-length decision
  - ...
  - Hardware Design
    - The performance improvements for steps



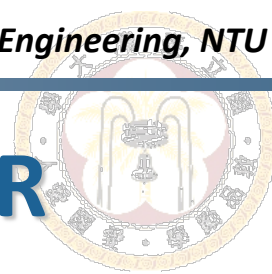
# Reference

- [1] Parallel High Throughput Soft-output Sphere Decoder : [Link](#)
- [2] [Video][Hung-yi Lee] Orthogonal Basis: Gram-Schmidt process : [Link](#)
- [3] Gram-Schmidt process : [Link](#)



# Appendix

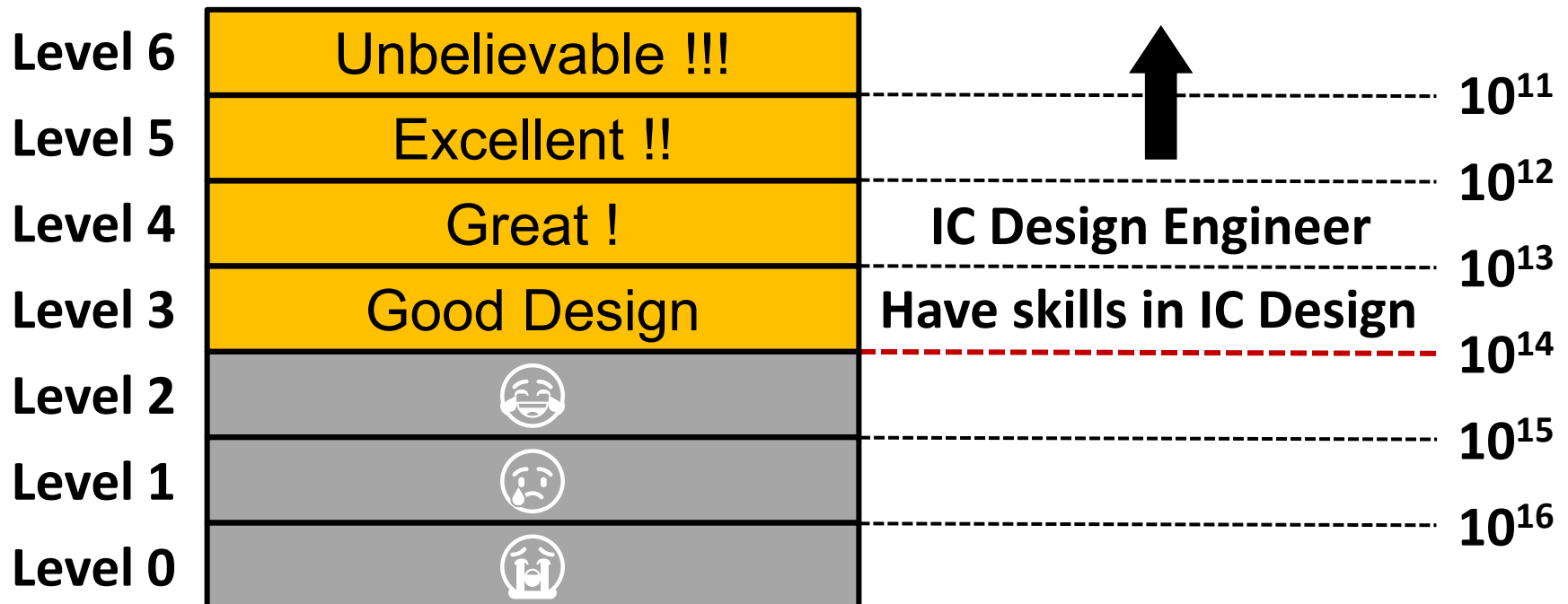


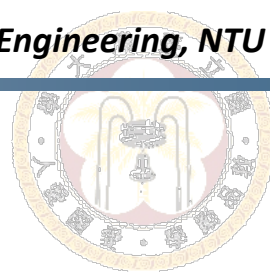


# Performance Evaluation Before APR

- Area ( $\mu\text{m}^2$ ): Total cell area (Synthesis)
- Time (ns): Simulation time of gatesim
- Power (mW): Total power of gatesim

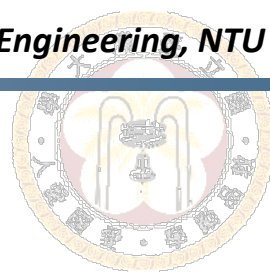
$$\frac{A \times T \times P}{\text{Performance Gain}}$$





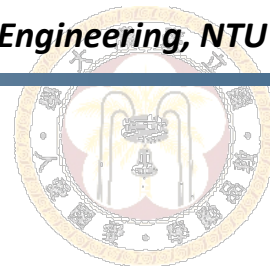
# Multiply

- 8 x 8 bits: 2.54 ns
- 12 x 12 bits: 3.11 ns
- 16 x 16 bits: 3.46 ns
- 20 x 20 bits: 3.83 ns
- 24 x 24 bits: 4.04 ns
- 28 x 28 bits: 4.23 ns
- 32 x 32 bits: 4.39 ns
- 36 x 36 bits: 4.63 ns



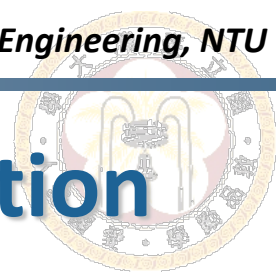
## DesignWare: Sqrt\_pipe

- Input 33 bits / Output 17 bits
- # of pipeline stage (1): 13.86 ns
- # of pipeline stage (2): 6.93 ns
- # of pipeline stage (3): 4.62 ns
- # of pipeline stage (4): 3.47 ns
- # of pipeline stage (5): 2.77 ns



## Soft LLR Error Rate

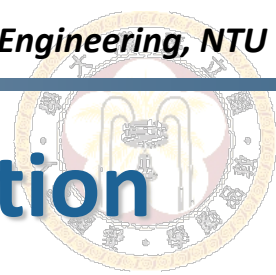
- $\text{sign}(\text{DUT Soft LLR}) == \text{sign}(\text{GLD Soft LLR}^{\text{note1}})$ 
  - $|llr_{dut}| \geq |llr_{gld}|$  : error rate = 0%
  - $|llr_{dut}| < |llr_{gld}|$  : error rate =  $(\text{abs}(llr_{gld}) - \text{abs}(llr_{dut})) / \text{abs}(llr_{gld})$
- $\text{sign}(\text{DUT Soft LLR}) \sim \text{sign}(\text{GLD Soft LLR})$ 
  - error rate =  $\text{abs}(llr_{dut} - llr_{gld}) / \text{abs}(llr_{gld})$



# Maximum Likelihood (ML) Demodulation

## ■ Formula

- LLR for  $x_{1,1} : L(x_{1,1}|\underline{y}) = \min_{\underline{x} \in X_{1,1,1}} \sum_{i=1}^4 |\hat{y}_i - \sum_{j=i}^4 R_{ij}s_j|^2 - \min_{\underline{x} \in X_{1,1,0}} \sum_{i=1}^4 |\hat{y}_i - \sum_{j=i}^4 R_{ij}s_j|^2$
- LLR for  $x_{1,2} : L(x_{1,2}|\underline{y}) = \min_{\underline{x} \in X_{1,2,1}} \sum_{i=1}^4 |\hat{y}_i - \sum_{j=i}^4 R_{ij}s_j|^2 - \min_{\underline{x} \in X_{1,2,0}} \sum_{i=1}^4 |\hat{y}_i - \sum_{j=i}^4 R_{ij}s_j|^2$
- LLR for  $x_{2,1} : L(x_{2,1}|\underline{y}) = \min_{\underline{x} \in X_{2,1,1}} \sum_{i=1}^4 |\hat{y}_i - \sum_{j=i}^4 R_{ij}s_j|^2 - \min_{\underline{x} \in X_{2,1,0}} \sum_{i=1}^4 |\hat{y}_i - \sum_{j=i}^4 R_{ij}s_j|^2$
- LLR for  $x_{2,2} : L(x_{2,2}|\underline{y}) = \min_{\underline{x} \in X_{2,2,1}} \sum_{i=1}^4 |\hat{y}_i - \sum_{j=i}^4 R_{ij}s_j|^2 - \min_{\underline{x} \in X_{2,2,0}} \sum_{i=1}^4 |\hat{y}_i - \sum_{j=i}^4 R_{ij}s_j|^2$
- LLR for  $x_{3,1} : L(x_{3,1}|\underline{y}) = \min_{\underline{x} \in X_{3,1,1}} \sum_{i=1}^4 |\hat{y}_i - \sum_{j=i}^4 R_{ij}s_j|^2 - \min_{\underline{x} \in X_{3,1,0}} \sum_{i=1}^4 |\hat{y}_i - \sum_{j=i}^4 R_{ij}s_j|^2$
- LLR for  $x_{3,2} : L(x_{3,2}|\underline{y}) = \min_{\underline{x} \in X_{3,2,1}} \sum_{i=1}^4 |\hat{y}_i - \sum_{j=i}^4 R_{ij}s_j|^2 - \min_{\underline{x} \in X_{3,2,0}} \sum_{i=1}^4 |\hat{y}_i - \sum_{j=i}^4 R_{ij}s_j|^2$
- LLR for  $x_{4,1} : L(x_{4,1}|\underline{y}) = \min_{\underline{x} \in X_{4,1,1}} \sum_{i=1}^4 |\hat{y}_i - \sum_{j=i}^4 R_{ij}s_j|^2 - \min_{\underline{x} \in X_{4,1,0}} \sum_{i=1}^4 |\hat{y}_i - \sum_{j=i}^4 R_{ij}s_j|^2$
- LLR for  $x_{4,2} : L(x_{4,2}|\underline{y}) = \min_{\underline{x} \in X_{4,2,1}} \sum_{i=1}^4 |\hat{y}_i - \sum_{j=i}^4 R_{ij}s_j|^2 - \min_{\underline{x} \in X_{4,2,0}} \sum_{i=1}^4 |\hat{y}_i - \sum_{j=i}^4 R_{ij}s_j|^2$



# Maximum Likelihood (ML) Demodulation

## ■ Formula

–  $s_1 \sim s_4$ : one of  $\left(\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}\mathbf{j}\right)$ ,  $\left(-\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}\mathbf{j}\right)$ ,  $\left(\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}}\mathbf{j}\right)$ , and  $\left(-\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}}\mathbf{j}\right)$

– At  $\sum_{i=1}^4 \left| \hat{y}_i - \sum_{j=i}^4 R_{ij} s_j \right|^2$  part:

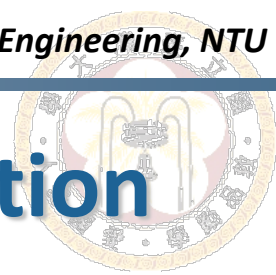
- 4<sup>th</sup> entry:  $\hat{y}_4 - R_{44}s_4 = a + bj \rightarrow a^2 + b^2$

- 3<sup>rd</sup> entry:  $\hat{y}_3 - R_{33}s_3 - R_{34}s_4 = c + dj \rightarrow c^2 + d^2$

- 2<sup>nd</sup> entry:  $\hat{y}_2 - R_{22}s_2 - R_{23}s_3 - R_{24}s_4 = e + fj \rightarrow e^2 + f^2$

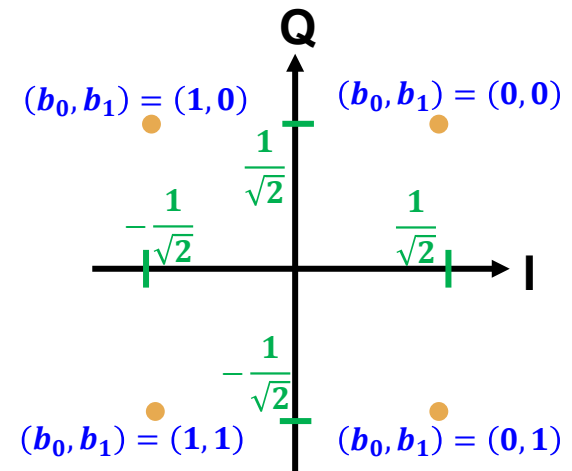
- 1<sup>st</sup> entry:  $\hat{y}_1 - R_{11}s_1 - R_{12}s_2 - R_{13}s_3 - R_{14}s_4 = g + hj \rightarrow g^2 + h^2$

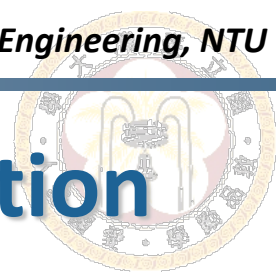
–  $\sum_{i=1}^4 \left| \hat{y}_i - \sum_{j=i}^4 R_{ij} s_j \right|^2 = a^2 + b^2 + c^2 + d^2 + e^2 + f^2 + g^2 + h^2$



# Maximum Likelihood (ML) Demodulation

- QPSK constellation
  - $\mathbf{x}_1 \sim \mathbf{x}_4$ : one of  $(0,0)$ ,  $(1,0)$ ,  $(0,1)$ , and  $(1,1)$
  - $\mathbf{s}_1 \sim \mathbf{s}_4$ : one of  $\left(\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}\mathbf{j}\right)$ ,  $\left(-\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}\mathbf{j}\right)$ ,  $\left(\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}}\mathbf{j}\right)$ , and  $\left(-\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}}\mathbf{j}\right)$
  - Total 256 ( $=4^4$ ) possibilities for 4-layer QPSK
  - **Full search with 256 possibilities**





# Maximum Likelihood (ML) Demodulation

- Compute  $\sum_{i=1}^4 \left| \hat{y}_i - \sum_{j=i}^4 R_{ij} s_j \right|^2$  for each A(path M), M=1~256
  - Bring in total **256 results** to compute each bit LLR, exactly **128 results** for  $\mathbf{X}_{k,b,0}$  and  $\mathbf{X}_{k,b,1}$  **without overlapping**

- A:** a set of all combinations of  $s_1$ - $s_4$  with total  $4^4 = 256$  combinations
- A(M):** one of the combination

