Contents

- **My introduction & skills**
  - Working experiences (*non-quantum parts*)
  - Coding skills (*non-quantum parts*)

- **Quantum error correction code**
  - Code construction based on stabilizer formalism

- **Quantum communication**
  - Quantum walk

DUC Manh Nguyen – Introduction

nguyenmanhduc18@gmail.com
+82-010-6617-1811

Education:

- University of Ulsan **(UOU)**, Korea, *02/2015-02/2020*
  - *Ph.D. degree*, school of electrical engineering
  - *Thesis:* quantum error correction code & quantum communication

- Ha-Noi university of science and technology **(HUST)**, Vietnam, *09/2007-06/2012*
  - *Bachelor degree*, school of electronic and telecommunication
  - *Thesis:* GSM telecommunication system

Highlights:

- 4+ years experiences in system on chip (SoC) design & SW engineering.
- 5 years experiences in academic research.
- High-level language simulation for algorithm and digital signal processing.
- Hardware description language, Test-bench simulation.
- FPGA emulation, FPGA porting, FPGA prototype.

SW languages:

- Verilog HDL, Shell.
- MATLAB, C/C++, Python.

Tools:

- RTL: NC-Verilog, ModelSim, Spyglass lint.
- Synthesis: Design complier.
- FPGA: Quartus (Altera), Xilinx ISE (Vivado).
- C/C++: Visual studio, GNU gcc.
- Python: Anaconda.
- MATLAB: MATLAB R2021a.
- Others: MAGMA for coding theory

OS:

- Window 10, Ubuntu LTS.
- Centos, Red Hat

Work experiences:

- **G2touch**, Korea, *03/2020-present*
  - Touchscreen controller IC design

- Advanced network system Vietnam **(ANSV)**, *06/2014-02/2015*
  - Network engineer

- Samsung Vietnam for mobile R&D center **(SVMC)**, *09/2012-05/2014*
  - SW engineer

- Panasonic R&D center Vietnam **(PRDCV),** *01/2012-03/2012*
  - Internship student.
  - 3D-Graphic software development.

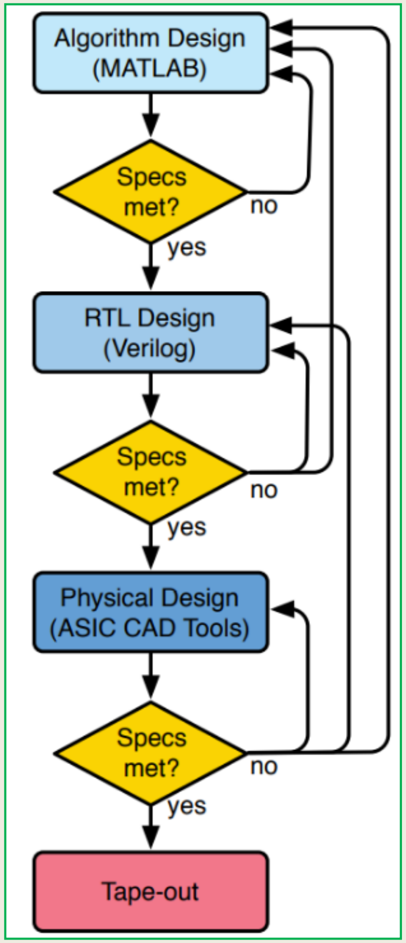
Languages:

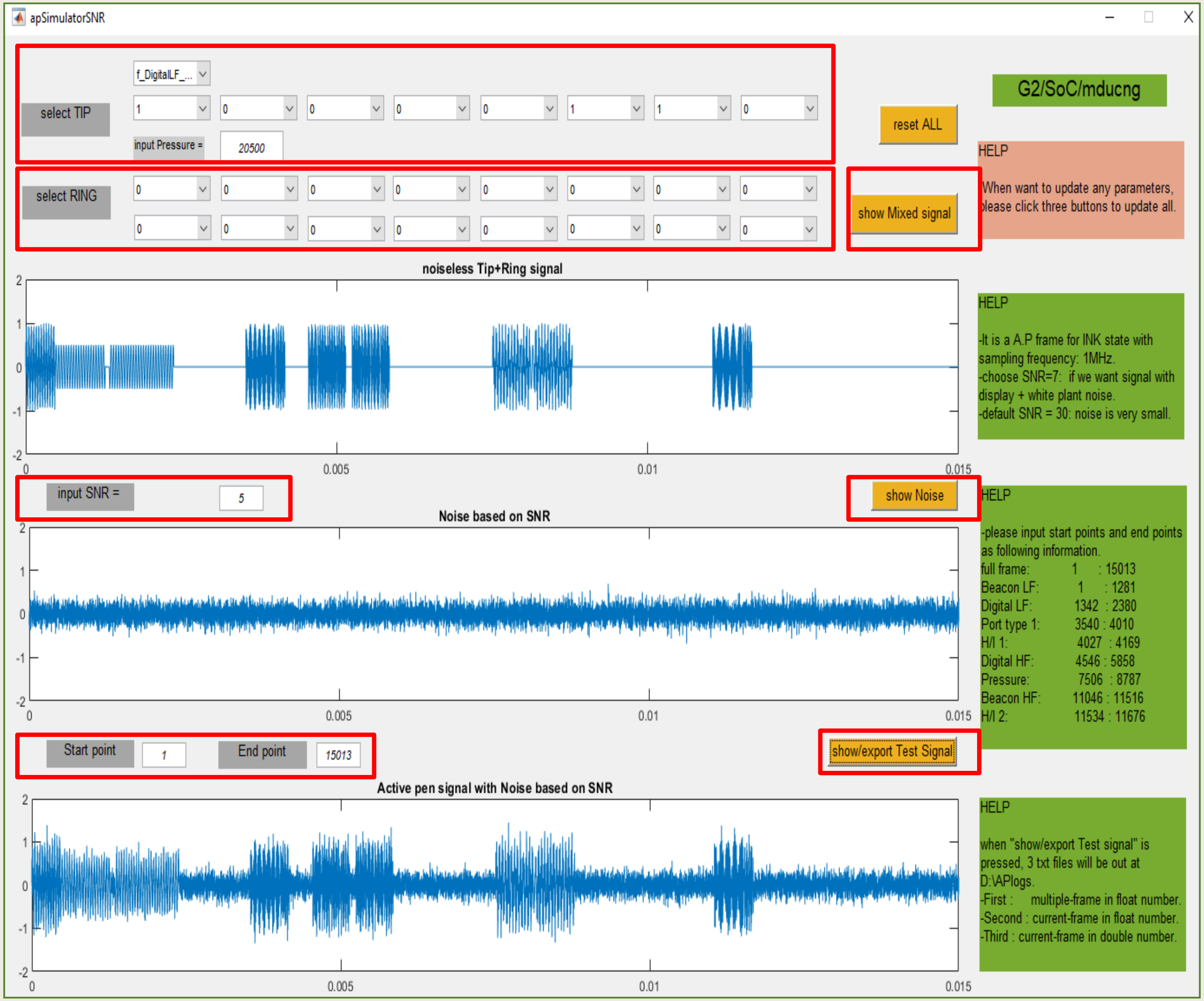
- Vietnamese (native)
- English (fluent)
- 한국어 (중급)

What I have done?

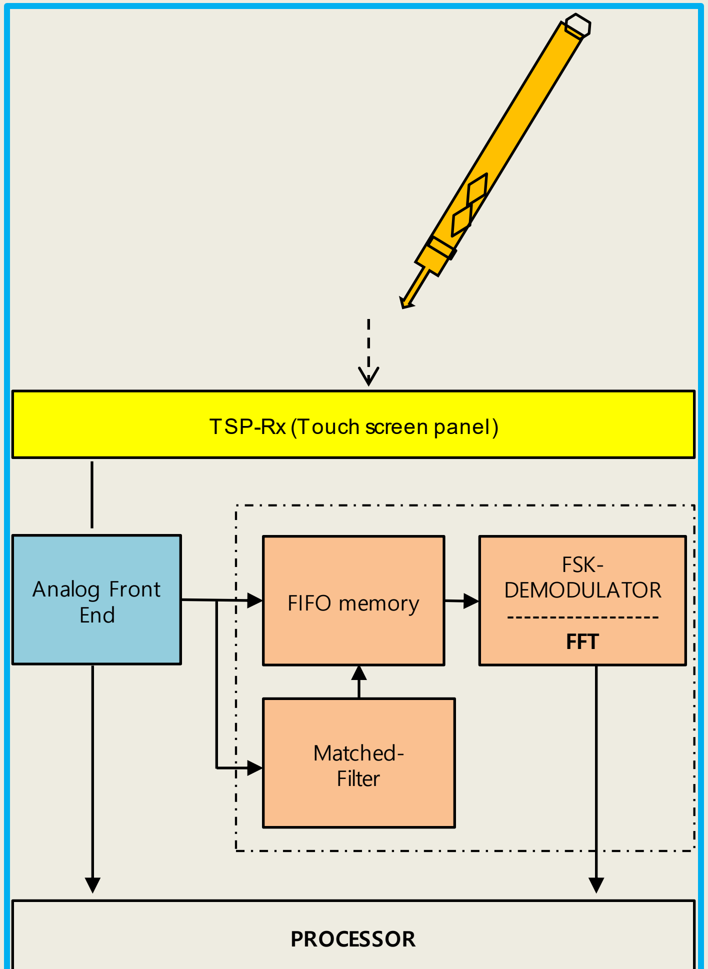
Project	Project descriptions	My roles	Year	Company
GL1T0AZ	.**LiDAR on chip** (light detection and ranging) for i ndustrial application and self-driving car.	.My proposal for G2touch new business idea. .VCSEL driver (with **DSSS sequence**) and SPAD detection (with **matched filter**) .TCSPC: Time-correlated single photon counting.	2021 ~	G2touch
GT7M0AZ	.**Touchscreen controller IC** with **USI** (universal stylus initiative) active pen support.	.**CRC** (Cyclic redundancy check) calculation logic .**DSSS** (Direct-sequence spread spectrum) logic generator .Pen signal decoder: **D-PSK** de-modulator .**FIR**, **IIR** digital filters design	2020 ~	
GT3T0A	.**Touchscreen controller IC** with **MPP** (Microsoft pen protocol) active pen support.	.**Matched filter** design for pen detector and synchronizer .Pen signal decoder: **FFT**, **M-FSK** .**SPI** bus interface test-bench simulation .**FPGA** based FTS (Finger touch sensing) module for verification .**Differential sensing** scheme based on **Gray-code** algorithm fo r display noise elimination. (*support analog team*)	2019-2021	
Papers	.Researches on **quantum information theory** an d **quantum algorithm**	.Quantum Error correction codes (**quantum stabilizer codes**) .Quantum communication: **Quantum walk**, quantum teleportatio n, quantum dense coding	2015 – 2020	UOU
.VMS phase 5	.Mobile **charging service** phase-V for MobiFone (t elecommunication system operator).	.**UNIX** Operating system .**Intelligent network platform**: SIGTRAN, SS7,… .**Mobile charging services** installation	2014-2015	ANSV
.S4-mini .S5-mini	.Galaxy smartphone S4-mini .Galaxy smartphone S5-mini	.SW binary **build set-up** .**Android software** modification for Galaxy phones	2012-2014	SVMC
.eCockpit	.Emergence of a **next generation UI** for automoti ve using 3D-GFX	.Implementation a 3D model (Hachune) using **ORGE 3D engine**	2 months – 2012	PRDCV

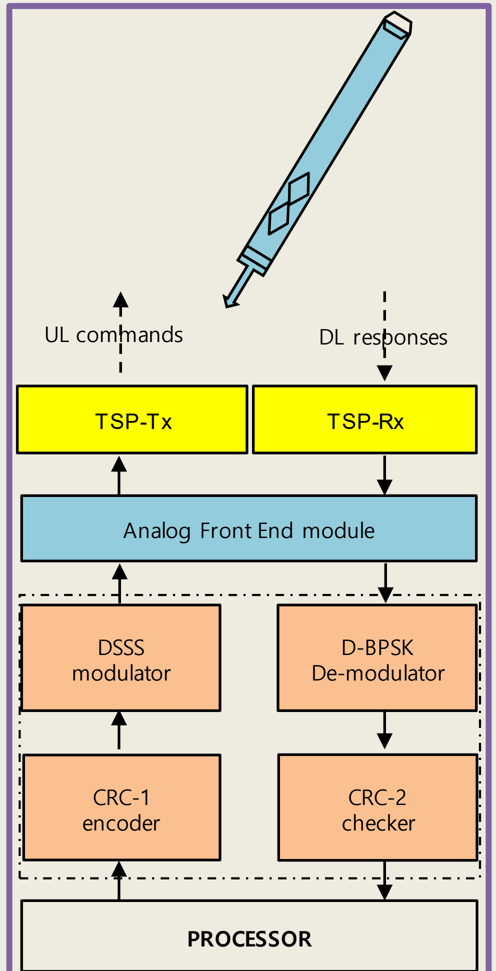
System and algorithm simulation: MATLAB





System and algorithm simulation: MATLAB





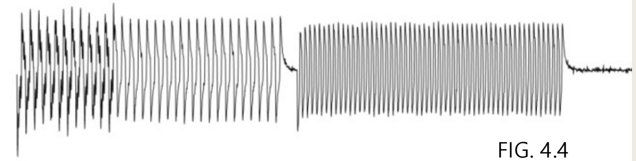


FIG. 4.4

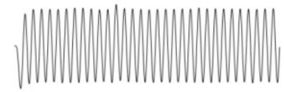


FIG. 4.5

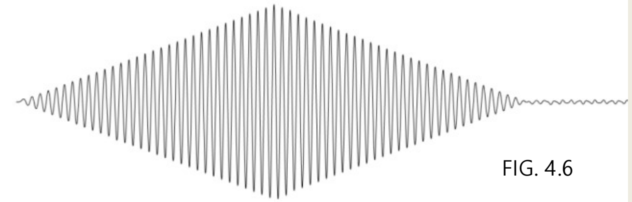


FIG. 4.6

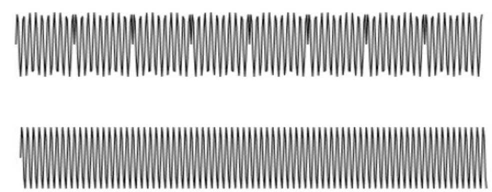


FIG. 6.2

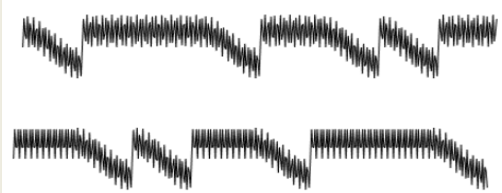


FIG. 6.3

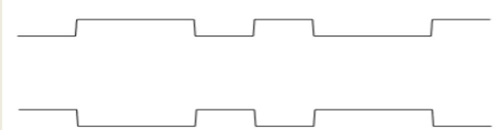
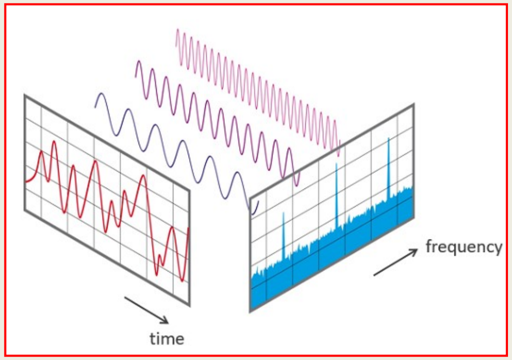
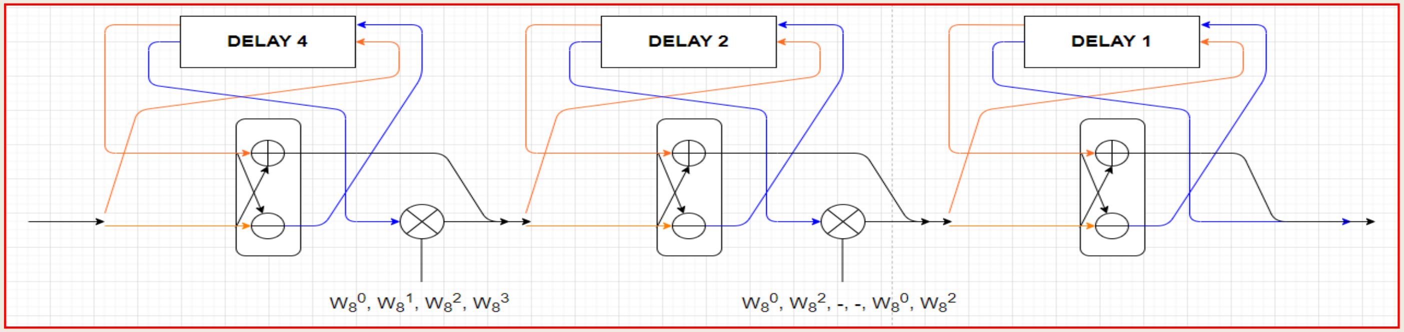
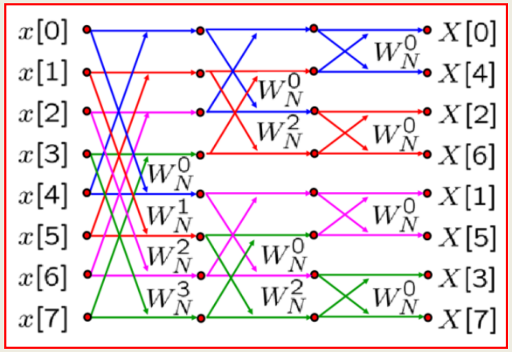


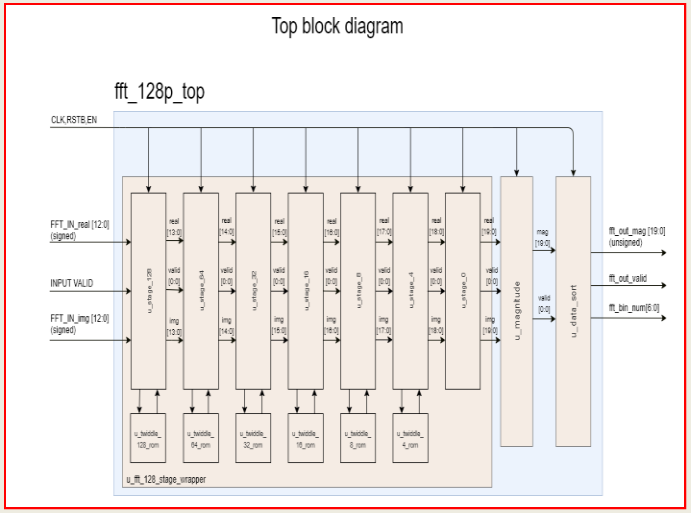
FIG. 6.4

Digital system design: Verilog HDL

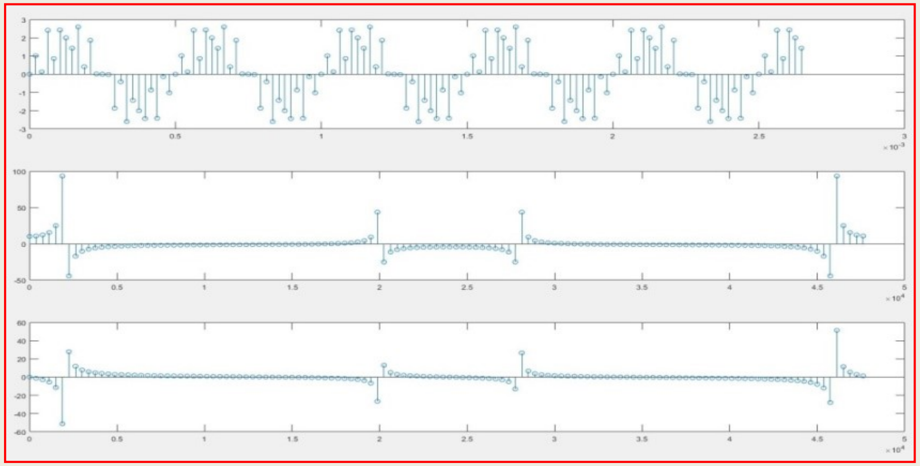


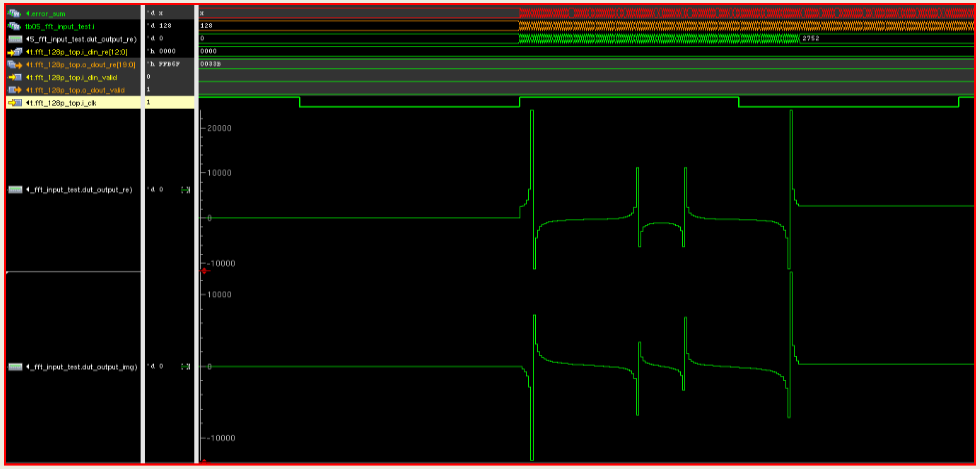






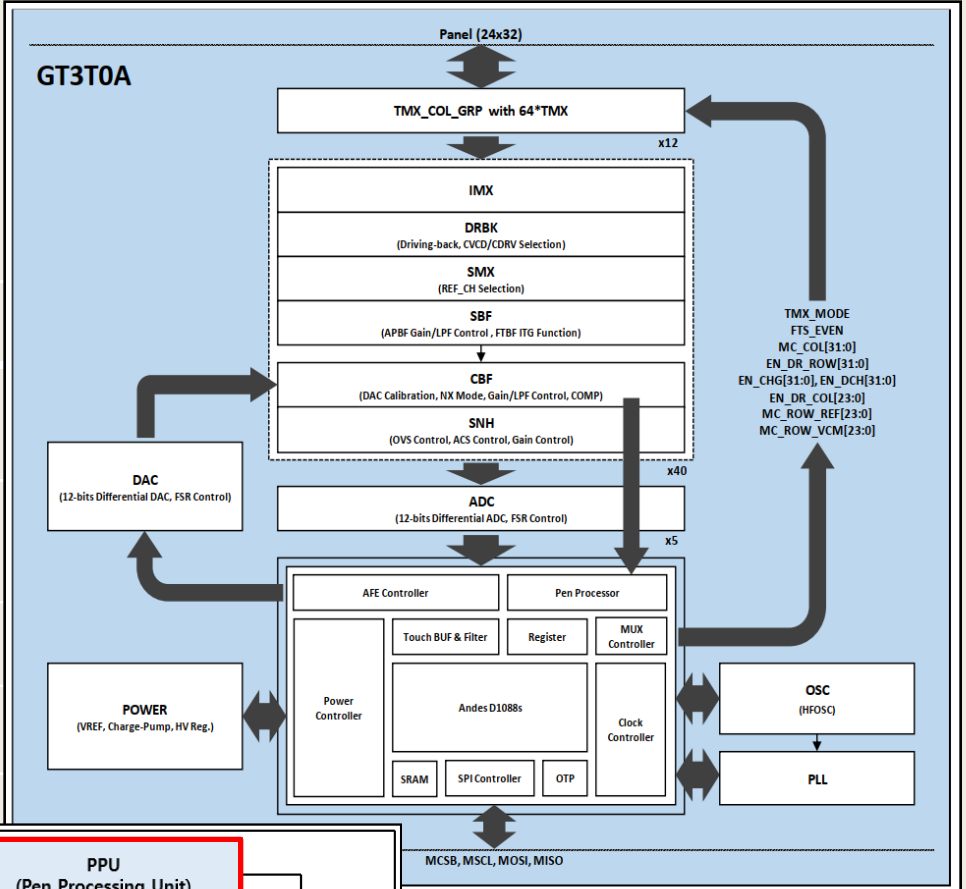

```
module fft_128p_top #(
        parameter IN_W = 13,             // input data bit width
        parameter OUT_W = 20,            // output data bit width
        parameter FFT_PT = 128,          // FFT point
        parameter FFT_PT_CNT_BIT = 7     // FFT point count bit width
) (
        input                            i_rst_b,
        input                            i_clk,
        input                            i_en,
        input   [IN_W-1:0]               i_din_re,
        input   [IN_W-1:0]               i_din_img,
        input                            i_din_valid,
        output [OUT_W-1:0]               o_dout_mag,
        output                           o_dout_valid,
        output [FFT_PT_CNT_BIT-1:0]      o_dout_bin_num
);
```



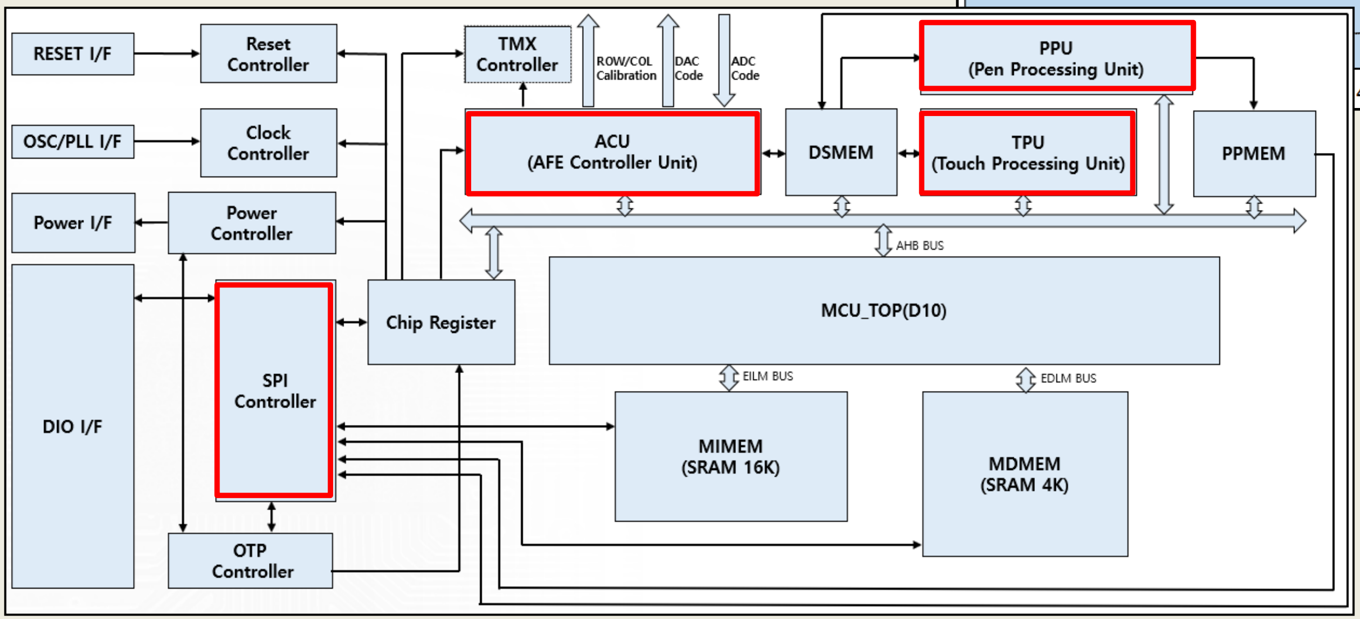


Digital system design: Verilog HDL

GT3T0A project	
Mixed IC:	.Analog + Digital signal processing .Digital system based on ARM architecture (ARM Cortex-M 3)
Application	.Touch sensing IC .MPP2.01 active pen → USI (Universal Stylus Initiative) active pen
Process:	.HV18GFL20 (0.18um HV 1-7M 1.8V/3.3V/20V technology)
End-user:	.SDC, BOE



4-1. GT3T0A Block Diagram



Software development: C++

Futuristic 3D UI Design

2011CES Audi Demo

App

2D/3D Animation Start

Ajax-CE VM

Upper Layer I/F for 3DUI

3D Control Lib

3D Rendering

3D Rendering Engine

2D Rendering

OpenGL

Extract the 3D Rendering Controller to Smoothly Shift to the use of HTML5

RGS, **OGRE**、Irrlicht Engine, Crystal Space, etc…

PRDCV's job !!

```
double vertices[] = {
    35.878300, 116.563200, 4.016700,    // 0
    31.7                                // 1
    38.1               -4.500900,       // 2
    34.045500, 95.141100, -3.019600,    // 3
    29.400300, 103.248500, -23.622200,  // 4
```

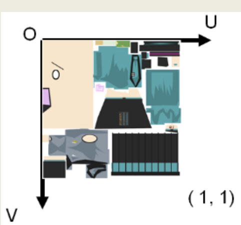
Points

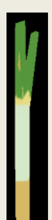
```
double indices[] = {
    8, 9, 10,     // 0
                  // 1
                  // 2
                  // 3
    48, 49, 50,   // 4
```

Triangles

```
double UVs[] = {
    0.984840, 0.090560,    // 0
                           // 1
                           // 2
                           // 3
    0.778350, 0.090560,    // 4
```

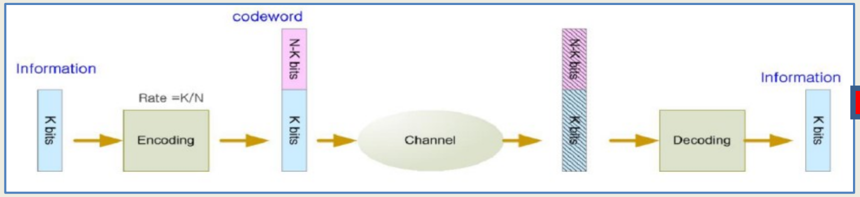
Points in 2D coordinate system

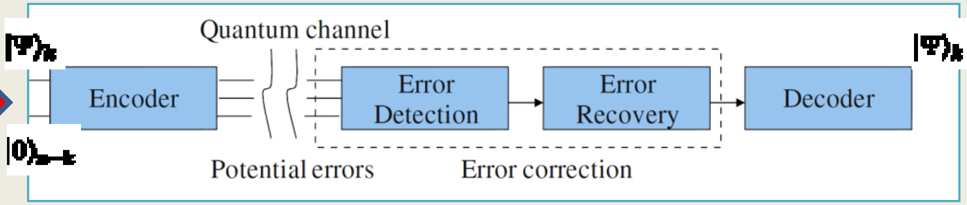




Quantum error correction code

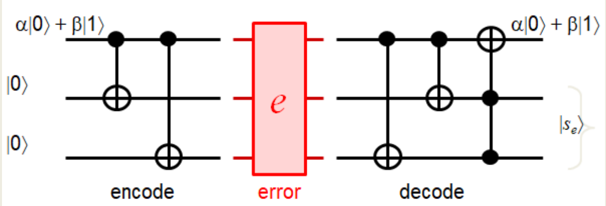
- Error correction



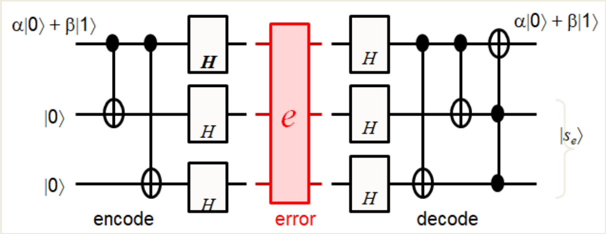


Potential errors: **X**, **Z**, **Y**

Repetition code

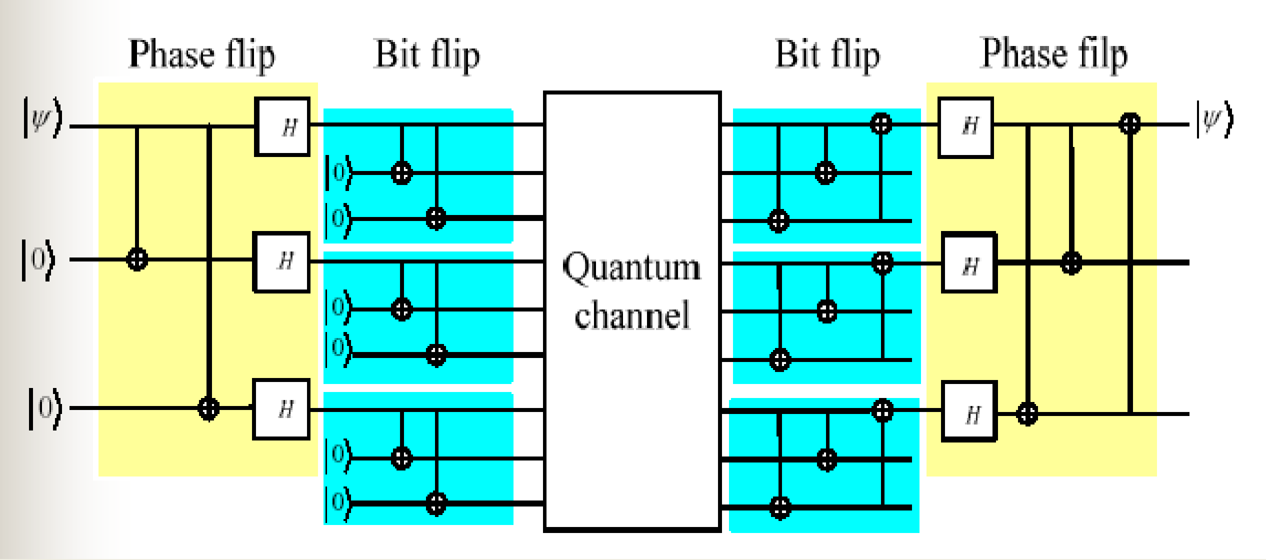


Bit flip (**X** error)



Phase flip (**Z** error)

Shor code



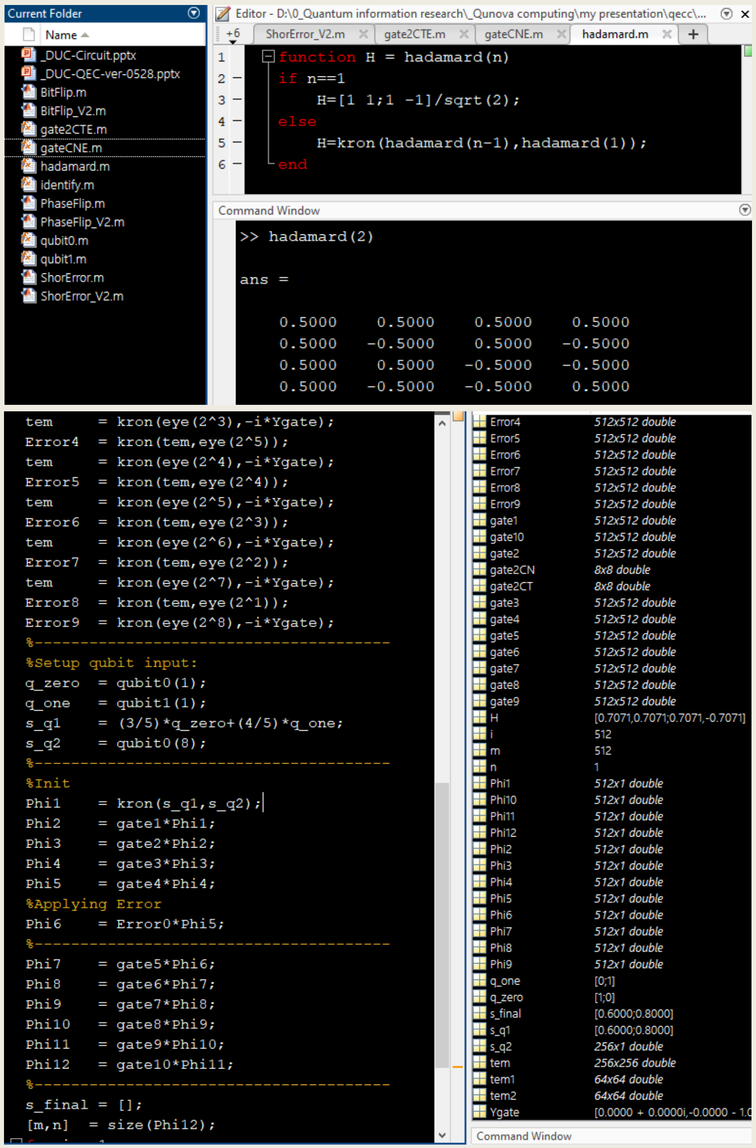
Bit & Phase flip (**Y** error)

Quantum error correction code

- Framework for simulation

SHOR CODE simulation

Based on MATLAB	Based on MAGMA




```
% Shor code:
F<i> := ComplexField(4);
H1   := HilbertSpace(F, 9);
f    := 3/5 * H1![0,0,0,0,0,0,0,0,0] + 4/5 * H1![1,0,0,0,0,0,0,0,0];
% Encoder:
ControlledNot(~f, {1}, 4);
ControlledNot(~f, {1}, 7);
f1   := BitFlip(f, 1);
f2   := PhaseFlip(f, 1);
f    := 1/SquareRoot(2)*f1 + 1/SquareRoot(2)*f2;
f1   := BitFlip(f, 4);
f2   := PhaseFlip(f, 4);
f    := 1/SquareRoot(2)*f1 + 1/SquareRoot(2)*f2;
f1   := BitFlip(f, 7);
f2   := PhaseFlip(f, 7);
f    := 1/SquareRoot(2)*f1 + 1/SquareRoot(2)*f2;
ControlledNot(~f, {1}, 2);
ControlledNot(~f, {1}, 3);
ControlledNot(~f, {4}, 5);
ControlledNot(~f, {4}, 6);
ControlledNot(~f, {7}, 8);
ControlledNot(~f, {7}, 8);
% Errors:
PhaseFlip(~f, 3);
BitFlip(~f, 3);
% Decoder:
ControlledNot(~f, {1}, 2);
ControlledNot(~f, {1}, 3);
ControlledNot(~f, {4}, 5);
ControlledNot(~f, {4}, 6);
ControlledNot(~f, {7}, 8);
ControlledNot(~f, {7}, 8);
ControlledNot(~f, {2,3}, 1);
ControlledNot(~f, {5,6}, 4);
ControlledNot(~f, {8,9}, 7);
f1   := BitFlip(f, 1);
f2   := PhaseFlip(f, 1);
f    := 1/SquareRoot(2)*f1 + 1/SquareRoot(2)*f2;
f1   := BitFlip(f, 4);
f2   := PhaseFlip(f, 4);
f    := 1/SquareRoot(2)*f1 + 1/SquareRoot(2)*f2;
f1   := BitFlip(f, 7);
f2   := PhaseFlip(f, 7);
f    := 1/SquareRoot(2)*f1 + 1/SquareRoot(2)*f2;
ControlledNot(~f, {1}, 4);
ControlledNot(~f, {1}, 7);
ControlledNot(~f, {4,7}, 1);
% Output:
f;
%http://magma.maths.usyd.edu.au/calc/
--> 0.5999|001100100> + 0.8000|101100100>
```

Quantum error correction code

- Stabilizer + Coding theory → Quantum stabilizer codes
  - Representing a state as its **group of stabilizers**

1. I stabilizes everything.
2. $-I$ stabilizes nothing.
3. X stabilizes $|+\rangle$: $X|+\rangle = \begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}\begin{bmatrix}\frac{1}{\sqrt{2}}\\\frac{1}{\sqrt{2}}\end{bmatrix} = \begin{bmatrix}\frac{1}{\sqrt{2}}\\\frac{1}{\sqrt{2}}\end{bmatrix} = |+\rangle$
4. $-X$ stabilizes $|-\rangle$: $-X|-\rangle = -\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}\begin{bmatrix}\frac{1}{\sqrt{2}}\\\frac{-1}{\sqrt{2}}\end{bmatrix} = \begin{bmatrix}\frac{1}{\sqrt{2}}\\\frac{-1}{\sqrt{2}}\end{bmatrix} = |-\rangle$
5. Y stabilizes $|+i\rangle$: $Y|+i\rangle = \begin{bmatrix}0 & -i\\i & 0\end{bmatrix}\begin{bmatrix}\frac{1}{\sqrt{2}}\\\frac{i}{\sqrt{2}}\end{bmatrix} = \begin{bmatrix}\frac{1}{\sqrt{2}}\\\frac{i}{\sqrt{2}}\end{bmatrix} = |+i\rangle$
6. $-Y$ stabilizes $|-i\rangle$: $-Y|-i\rangle = -\begin{bmatrix}0 & -i\\i & 0\end{bmatrix}\begin{bmatrix}\frac{1}{\sqrt{2}}\\\frac{-i}{\sqrt{2}}\end{bmatrix} = \begin{bmatrix}\frac{1}{\sqrt{2}}\\\frac{-i}{\sqrt{2}}\end{bmatrix} = |-i\rangle$
7. Z stabilizes $|0\rangle$: $Z|0\rangle = \begin{bmatrix}1 & 0\\0 & -1\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}1\\0\end{bmatrix} = |0\rangle$
8. $-Z$ stabilizes $|1\rangle$: $-Z|1\rangle = -\begin{bmatrix}1 & 0\\0 & -1\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}0\\1\end{bmatrix} = |1\rangle$

Critical result from group theory: for any N-qubit stabilized state, only N elements needed to specify group.

1. $|0\rangle$ is stabilized by $\{I, Z\}$, Z is generator
2. $|1\rangle$ is stabilized by $\{I, -Z\}$, -Z is generator
3. $|+\rangle$ is stabilized by $\{I, X\}$, X is generator
4. $|-\rangle$ is stabilized by $\{I, -X\}$, -X is generator
5. $|+i\rangle$ is stabilized by $\{I, Y\}$, Y is generator
6. $|-i\rangle$ is stabilized by $\{I, -Y\}$, -Y is generator
7. $|0\rangle \otimes |0\rangle$ is stabilized by $\{I\otimes I, I\otimes Z, Z\otimes I, Z\otimes Z\}$, $\{I\otimes Z, Z\otimes I\}$ is generator
8. $|+\rangle \otimes |0\rangle$ is stabilized by $\{I\otimes I, I\otimes Z, X\otimes I, X\otimes Z\}$, $\{I\otimes Z, X\otimes I\}$ is generator
9. $\frac{|00\rangle+|11\rangle}{\sqrt{2}}$ is stabilized by $\{I\otimes I, X\otimes X, -Y\otimes Y, X\otimes Z\}$, $\{X\otimes X, Z\otimes Z\}$ is generator

$$\mathbf{E}|\Psi\rangle = \mathbf{E}\mathbf{M}_i|\Psi\rangle = \begin{cases}\mathbf{M}_i\mathbf{E}|\Psi\rangle & \text{Error undetected}\\ -\mathbf{M}_i\mathbf{E}|\Psi\rangle & \text{Error detected}\end{cases}$$

$[[n,k,d]]$

$$\sim \begin{pmatrix} Z & Z & X & I & X\\ X & Z & Z & X & I\\ I & X & Z & Z & X\\ X & I & X & Z & Z \end{pmatrix}$$

$$\text{e.g.}\quad H = \left(\begin{array}{ccccc|ccccc} 1&1&0&0&0&0&0&1&0&1\\ 0&1&1&0&0&1&0&0&1&0\\ 0&0&1&1&0&0&1&0&0&1\\ 0&0&0&1&1&1&0&1&0&0 \end{array}\right)$$

Pauli operator	$GF(4)$ elements
I	0
X	1
Y	$\omega^2 = \omega + 1$
Z	ω
Multiplication operator	Addition
Commutative	Trace-inner product

→

Symplectic inner product

$$\mathbf{M}_i\mathbf{M}_j = \mathbf{M}_j\mathbf{M}_i$$

$$H(i,:)\odot H(j,:) = 0$$

$$\mathbf{H_X}.\mathbf{H_Z}^T + \mathbf{H_Z}.\mathbf{H_X}^T = 0$$

Quantum error correction code

- My main contribution: $\boxed{\text{symplectic product } (u,v)\odot(\pi^{x}(u),\pi^{-x}(v)) \text{ is zero}}$

$$
\mathbf{H}=\left[\begin{array}{ccccccc|ccccccc}
1&1&0&0&1&0&1&1&0&0&1&0&1&1\\
1&1&1&0&0&1&0&0&0&1&0&1&1&1\\
0&1&1&1&0&0&1&0&1&0&1&1&1&0\\
1&0&1&1&1&0&0&1&0&1&1&1&0&0\\
0&1&0&1&1&1&0&0&1&1&1&0&0&1\\
0&0&1&0&1&1&1&1&1&1&0&0&1&0\\
1&0&0&1&0&1&1&1&1&0&0&1&0&1
\end{array}\right]
$$

$$
\begin{array}{cccccc}
\scriptstyle r & \scriptstyle n-k-r & \scriptstyle k & \scriptstyle r & \scriptstyle n-k-r & \scriptstyle k
\end{array}
$$
$$
\left[\begin{array}{cccccc}
\mathbf{I} & \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{B} & \mathbf{C}_1 & \mathbf{C}_2\\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{D} & \mathbf{I} & \mathbf{E}
\end{array}\right]\begin{array}{l}\}\; r\\ \}\; n-k-r\end{array}
$$

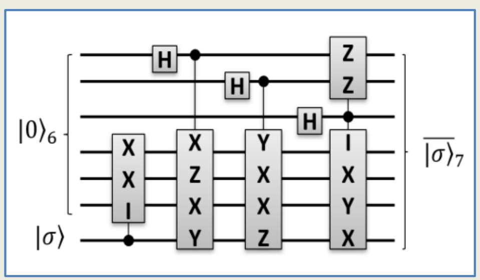
$$
\begin{cases}
\overline{\mathbf{X}}=\left[\mathbf{0}\;\; \mathbf{E}^T\;\; \mathbf{I}\;\; (\mathbf{E}^T\mathbf{C}_1+\mathbf{C}_2{}^T)\;\; \mathbf{0}\;\; \mathbf{0}\right]\\
\overline{\mathbf{Z}}=\left[\mathbf{0}\;\; \mathbf{0}\;\; \mathbf{0}\;\; \mathbf{A}_2{}^T\;\; \mathbf{0}\;\; \mathbf{I}\right]
\end{cases}
$$

$$
\left|\overline{c_1c_2\ldots c_k}\right\rangle=\frac{1}{\sqrt{2^m}}\times\left(\prod_{i=1}^{m}(\mathbf{I}+\mathbf{g}_i)\right)\times\overline{\mathbf{X}_1}^{c_1}\times\overline{\mathbf{X}_2}^{c_2}\times\ldots\times\overline{\mathbf{X}_k}^{c_k}\left|00\ldots0\right\rangle_n,
$$

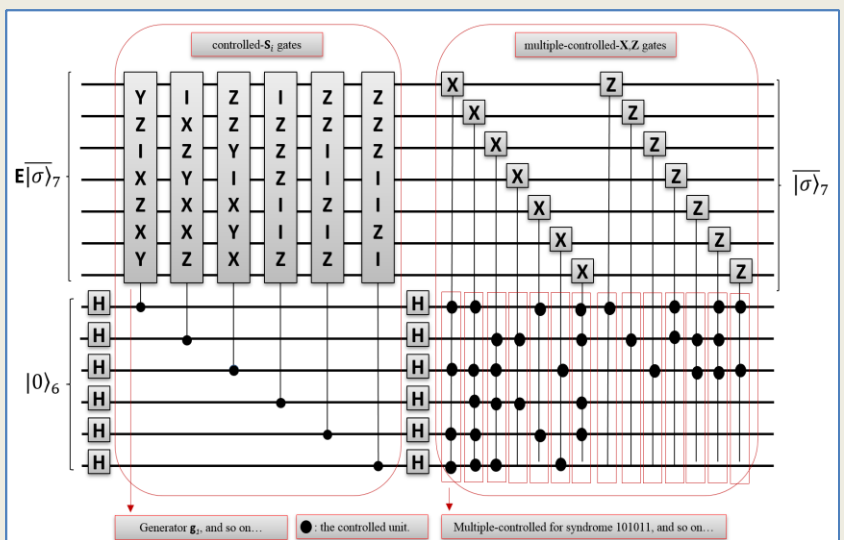
$$
\begin{aligned}
\left|\overline{\sigma_1\sigma_2\ldots\sigma_k}\right\rangle&=\frac{1}{\sqrt{2^{n-k}}}\times\left(\prod_{i=1}^{n-k}(\mathbf{I}+\mathbf{g}_i)\right)\times\overline{\mathbf{X}_1}^{\sigma_1}\times\overline{\mathbf{X}_2}^{\sigma_2}\times\ldots\times\overline{\mathbf{X}_k}^{\sigma_k}\left|00\ldots0\right\rangle_n\\
&=\left(\prod_{i=1}^{r}T_i^{\sigma_i}\mathbf{H}_i\right)\times\left(\prod_{l=1}^{k}\mathbf{U}_l\right)\left|00\ldots0\sigma_1\sigma_2\ldots\sigma_k\right\rangle_n.
\end{aligned}
$$

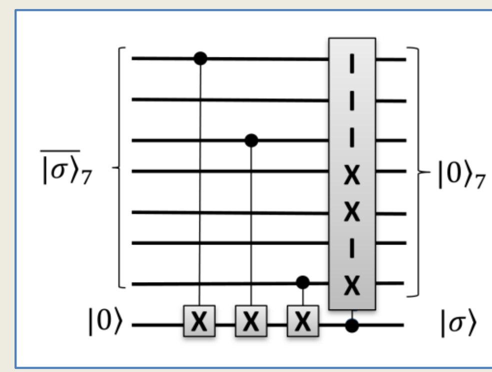
$$
\mathbf{g}_i\left|\psi\right\rangle=(-1)^{l_i}\left|\psi\right\rangle\Leftrightarrow\mathbf{g}_i\mathbf{E}=(-1)^{l_i}\mathbf{E}\mathbf{g}_i
$$

$$
\left|\psi_{in}\right\rangle=\left|\overline{\sigma_1\sigma_2\ldots\sigma_k}\right\rangle_n\otimes\left|00\ldots0\right\rangle_k
$$
$$
\left|\psi_f\right\rangle=\mathbf{U}_{decode}\left|\overline{\sigma_1\sigma_2\ldots\sigma_k}\right\rangle_n\otimes\left|00\ldots0\right\rangle_k=\left|\overline{00\ldots0}\right\rangle_n\otimes\left|\sigma_1\sigma_2\ldots\sigma_k\right\rangle_k.
$$



E	Syndrome	E	Syndrome	E	Syndrome
XIIIIII	101011	ZIIIIII	100000	YIIIIII	001011
IXIIIII	101111	IZIIIII	010000	IYIIIII	111111
IIXIIII	011101	IIZIIII	001000	IIYIIII	010101
IIIXIII	010100	IIIZIII	110000	IIIYIII	100100
IIIIXII	100010	IIIIZII	011000	IIIIYII	111010
IIIIIXI	001001	IIIIIZI	111000	IIIIIYI	110001
IIIIIIX	110110	IIIIIIZ	101000	IIIIIIY	011110





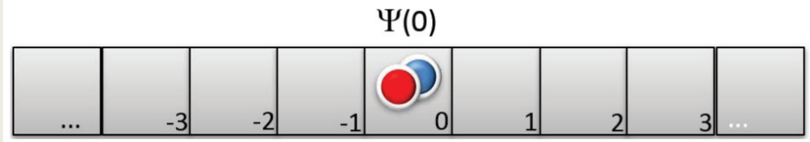
Quantum communication: Quantum walk

A discrete-time QWs over the line involves in two Hilbert spaces. The total space of the walk is given as

$$H = H_P \otimes H_C, \tag{1}$$

where H_P is the position space, which is spanned by $\{|n\rangle \text{ where } n \in \mathrm{Z}\}$, and H_C is the coin space, which is spanned by $\{|0\rangle, |1\rangle\}$.

$$\mathbf{S}^T = \left[\sum_{-\infty}^{+\infty} |n+1\rangle\langle n| \otimes |0\rangle\langle 0| + \sum_{-\infty}^{+\infty} |n-1\rangle\langle n| \otimes |1\rangle\langle 1|\right]^T$$
$$= \sum_{-\infty}^{+\infty} |n\rangle\langle n+1| \otimes |0\rangle\langle 0| + \sum_{-\infty}^{+\infty} |n\rangle\langle n-1| \otimes |1\rangle\langle 1|.$$



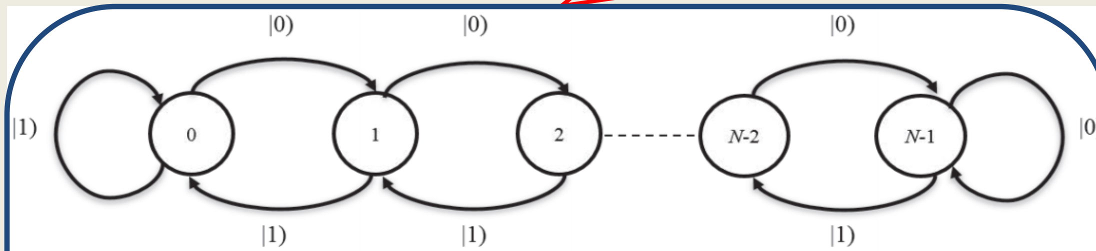


Fig. 2. The QW over the N-line.

$$\mathbf{S}_1 = \sum_{t=0}^{N-2} |t+1\rangle\langle t| \otimes |0\rangle\langle 0| + |N-1\rangle\langle N-1| \otimes |0\rangle\langle 0|$$
$$+ \sum_{t=1}^{N-1} |t-1\rangle\langle t| \otimes |1\rangle\langle 1| + |0\rangle\langle 0| \otimes |1\rangle\langle 1|.$$

$$\mathbf{S}_1 * \mathbf{S}_1^T = \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \cdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \cdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \end{bmatrix} \neq \mathbf{I}.$$

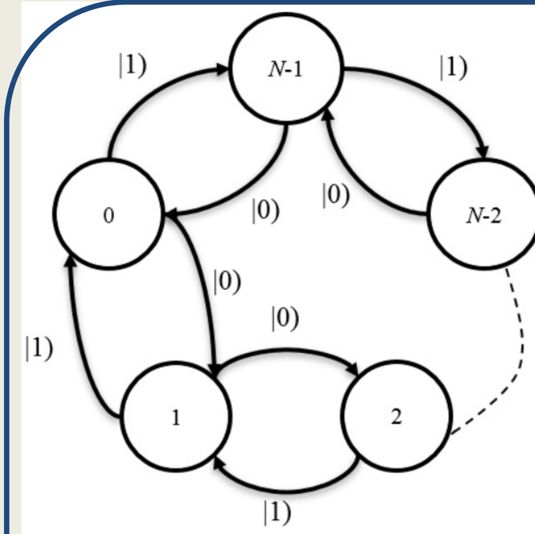
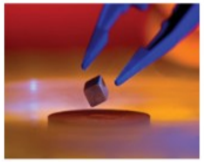


Fig. 3. The QW over N-cycle.

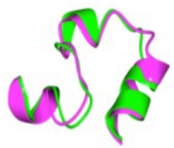
$$\mathbf{S}_2 = \sum_{t=0}^{N-2} |t \oplus 1\rangle\langle t| \otimes |0\rangle\langle 0| + \sum_{t=1}^{N-1} |t \ominus 1\rangle\langle t| \otimes |1\rangle\langle 1|,$$

$$\mathbf{S}_2 * \mathbf{S}_2^\dagger = \mathbf{S}_2 * \mathbf{S}_2^T = \mathbf{I},$$
$$\mathbf{S}_2^\dagger * \mathbf{S}_2 = \mathbf{S}_2^T * \mathbf{S}_2 = \mathbf{I}.$$

Plan for working in Qunova computing



Energy
Room-temperature superconductivity



Health
Quantum chemistry

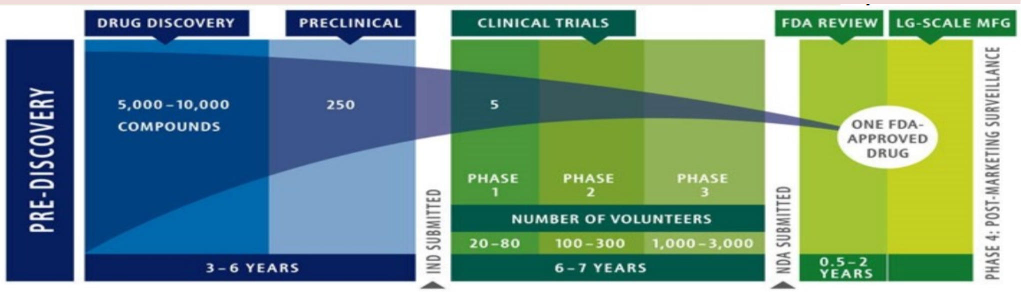
Internet Security

A computation is a physical process. It may be performed by a piece of electronics or on an abacus, or in your brain, but it is a process that takes place in nature and as such it is subject to the laws of physics.
Quantum computers are machines that rely on characteristically quantum phenomena, such as quantum interference and quantum entanglement in order to perform computation.

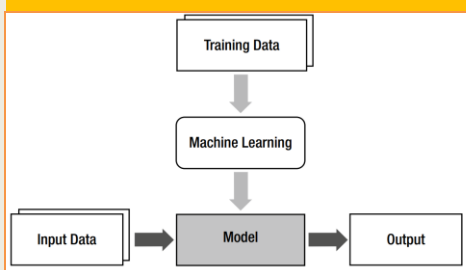
– Artur Ekert

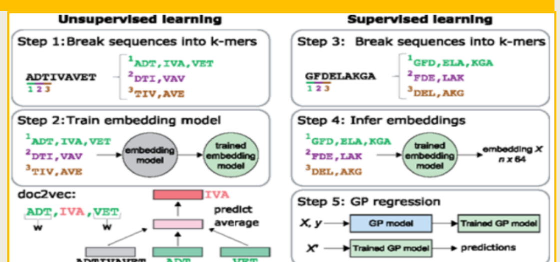
Overarching goal
Solve intractable problems with massive speedup in computation…
superposition and entanglement, two of the cornerstones
echanics

Drug discovery process



ML/ AI





Quantum computing

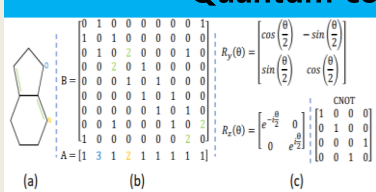


FIGURE 2. (a-b) A sample molecular graph from QM9 denoted by its corresponding atom vector A and bond matrix B; (c) all quantum gates used in this study.

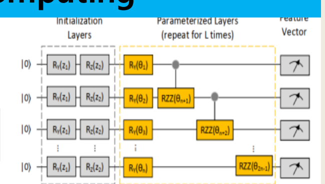
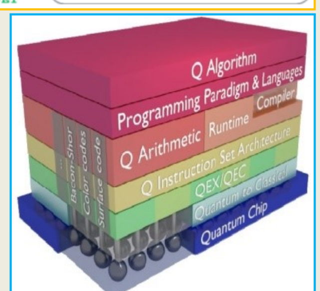
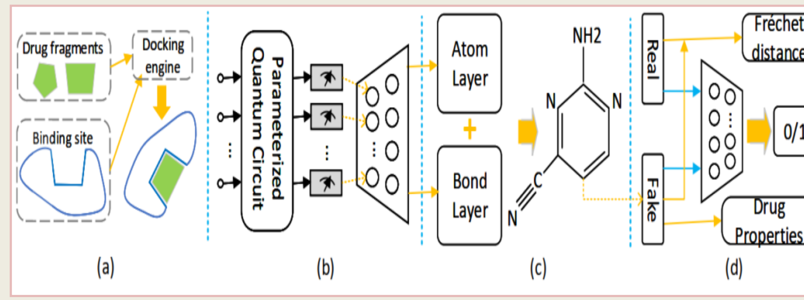


FIGURE 3. Parameterized quantum circuit to obtain feature vector of N dimensions. The circuit is composed of initialization layers, repeatable parameterized layers and measurement layer. Each RZZ gate is composed of two CNOT gates and one parametric RZ gate.





-Using quantum computing with **parameterized circuit** to boost and generate **novel drug patterns.**