درس پانزدهم: کدهای اصلاح کننده خطا درکامپیوترهای کوانتومی

۱ مقدمه

وقتی که بیت های کلاسیک از یک کانال کلاسیک عبور می کنند همواره ممکن است دچار خطا شوند. این خطاها هنگام پردازش اطلاعات در یک کامپیوتر کلاسیک نیز می توانند رخ دهند. بنابراین لفظ کانال دراین جا به یک معنای عام به کار می رود که الزاماً به معنای آن نیست که بیت یک فاصله مکانی طولانی را طی می کند. این اتفاق برای کیوبیت ها نیز می توانند رخ دهد یعنی کیوبیت ها نیز وقتی که از یک کانال کوانتومی عبور می کنند ممکن است دچار خطا شوند. یک کامپیوتر کوانتومی وقتی می تواند به خوبی کار کند که بتوانیم خطاهای ایجاد شده در کیوبیت ها را آشکار ساخته و تصحیح کنیم. این امر هم چنین برای ارسال اطلاعات کوانتومی توسط کیوبیت ها ضرورت دارد. از این به بعد لفظ کانال کلاسیک یا کانال کوانتومی را همواره به یک معنای عام به کار می بریم. چند تفاوت مهم بین خطاهای کلاسیک و کوانتومی وجود دارد و به نظر می رسد که خطاهای کوانتومی را واقعاً نمی توان آشکار و تصحیح کرد:

۱ — یک بیت کلاسیک می تواند ولتاژ یک نقطه و یا جریان عبور کننده از یک نقطه و یا مغناطش یک حوزه مغناطیسی باشد و اگر چه از نظر ماکروسکوپی کوچک است ولی از نظر میکروسکوپی هنوز از میلیارد ها اتم تشکیل شده است و امکان بروز خطا در آن بسیار کم است . در حالیکه یک کیوبیت واقعاً میکروسکوپی است و نشان دهنده حالت یک اتم یا اسپین یا یون است و براحتی می تواند دچار خطا شود.

۲ — خطای ایجاد شده در یک بیت کلاسیک یک خطای گسسته است که در آن مقدار $0(1)$ به مقدار $1(0)$ تبدیل می شود و حال آنکه خطای ایجاد شده در یک کیوبیت پیوسته است و بردار حالت بیت کوانتومی می تواند به طورپیوسته تغییر کند.

۳ — می توان یک بیت کلاسیک را مشاهده کرد و از خطای ایجاد شده در آن آگاهی یافت ولی یک بیت کوانتومی را براحتی نمی توان مشاهده کرد زیرا مشاهده‌ی آن عموماً منجر به کاهش تابع حالت و از بین رفتن حالت اولیه می شود.

۴ — و بالاخره می توان نسخه‌های متعددی از یک بیت کلاسیک را تهیه کرد که در صورت بروز خطا در یکی یا چند تا از آنها با استفاده از قانون اکثریت بتوان به بروز خطا در آن ها پی برد و حال آنکه بنابر قضیه عدم تکثیر *No cloning theorem* نمی توان یک حالت کوانتومی را تکثیر کرد و نسخه های متعددی از آن درست کرد.

تفاوت های بالا یا به عبارت بهتر موانع بالا در سالهای اولیه طرح نظریه کامپیوتر ها و اطلاعات کوانتومی باعث شده بود که عده‌ای فکر کنند کامپیوترهای کوانتومی یا مخابره اطلاعات کوانتومی هرگز به عمل نزدیک نخواهد شد زیر هرگز نمی توان خطاهای کوانتومی را تصحیح کرد. خوشبختانه امروزه یک نظریه کامل برای آشکارسازی و تصحیح خطاهای کوانتومی تدوین شده است که می تواند بر تمام موانع بالا غلبه کند و محاسبه و مخابره اطلاعات کوانتومی را به طوری که از خطا مصون باشد، امکان پذیر کند. هدف ما در این درس آن است که با این نظریه آشنا شویم. در این درس نخست مقدمه ای درباره روش های تصحیح خطاهای کلاسیک خواهیم گفت و سپس به تصحیح خطاهای کوانتومی خواهیم پرداخت. باید تاکید کنیم که بحث ما درباره نظریه تصحیح خطاهای کلاسیک یا به اصطلاح کدهای تصحیح کننده خطاهای کلاسیک بسیار مقدماتی است و تنها برای آن ارایه می شود که مقدمه ای برای نظریه کوانتومی باشد که روش های خود را از آن به عاریت گرفته است.

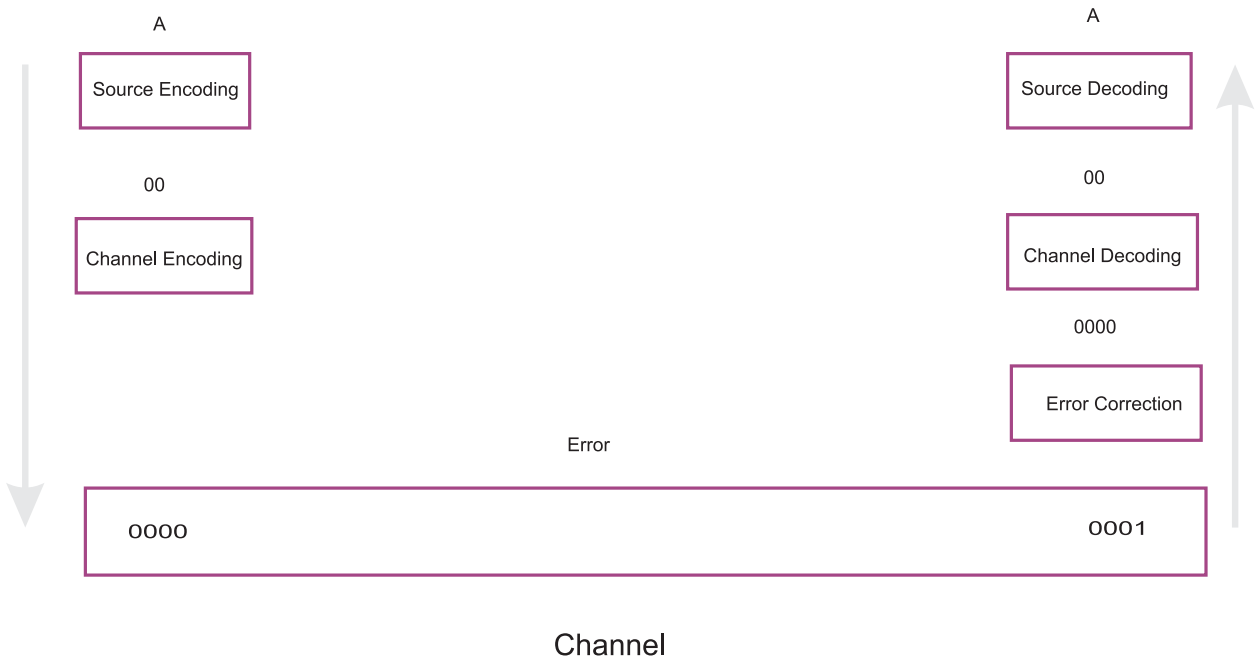
۲ اصلاح خطاهای کلاسیک

می دانیم که هرنوع اطلاعات کلاسیک به صورت رشته ای از علائم 0 و 1 کُد کرد. این نوع کُد کردن اصطلاحاً کُد کردن منبع یا *Source Coding* نامیده می شود و قبل از ارسال اطلاعات به درون کانال انجام می شود و هدف آن تنها این است که اطلاعاتی که به صورت متن، صدا یا تصویر است به صورت رشته ای از علائم ساده و قابل حمل دربیت های کلاسیک درآید. در کُد کردن منبع مسئله‌ای به نام تصحیح خطا وجود ندارد زیرا هنوز اطلاعات به درون کانال که جایی است که خطاها در آن صورت می گیرد، ارسال نشده‌است. به عنوان مثال فرض کنید که منبع ما تنها از چهار حرف A ، B ، C و D تشکیل شده است. یک راه برای کد کردن منبع آن است که این حروف را به رشته های زیر کُد کنیم:

$$\begin{aligned} A &\longrightarrow 00 \\ B &\longrightarrow 01 \\ C &\longrightarrow 10 \\ D &\longrightarrow 11. \end{aligned} \tag{1}$$

در مقصد نیز وقتی که این رشته ها به دست گیرنده می رسد او نیز همین روش را برای گشودن کد به کار می برد که به آن *Source Decoding* می گوییم. در حالت ساده بالا این عمل به صورت زیر انجام می شود:

$$\begin{aligned} 00 &\longrightarrow A \\ 01 &\longrightarrow B \\ 10 &\longrightarrow C \\ 11 &\longrightarrow D. \end{aligned} \tag{2}$$



شکل ۱: مراحل مختلف کد کردن و گشودن یک پیام

اما وقتی که همین رشته ها را به درون کانال می فرستیم می توانند دچار خطا شوند. به عنوان مثال رشته 01 می تواند دراثر بروز یک خطا در بیت اول تبدیل به 11 شود که در مقصد به صورت حرف D تعبیر خواهد شد. بنابراین برای تصحیح این گونه خطاها که در کانال اتفاق می افتد می بایست در ابتدای کانال یک نوع کد گذاری به کار برد که به آن کدگذاری کانال $ChannelEncoding$ می گوییم . به عنوان مثال یک نوع ساده از کد گذاری کانال آن است که رشته های چهارگانه فوق را به صورت زیر کد کنیم:

$$\begin{aligned} 00 &\longrightarrow 0000 \\ 01 &\longrightarrow 0101 \\ 10 &\longrightarrow 1010 \\ 11 &\longrightarrow 1111. \end{aligned} \tag{3}$$

در گیرنده نیز می بایست این رشته ها را به صورت زیر بگشاییم که به آن $ChannelDecoding$ می گوییم:

$$\begin{aligned} 0000 &\longrightarrow 00 \\ 0101 &\longrightarrow 01 \\ 1010 &\longrightarrow 10 \\ 1111 &\longrightarrow 11. \end{aligned} \tag{4}$$

مراحل چهارگانه فوق در شکل ۱ نشان داده شده است.

با این مقدمه می توانیم تعریف کلی کدهای کلاسیک را ارائه دهیم. قبل از آن می بایست نماد گذاری خود را روشن کنیم.

تعریف: مجموعه $Z_2^n := Z_2 \times Z_2 \times \cdots Z_2$ عبارت است از مجموعه تمام n تایی های مرتب که عناصر آن از 0 و 1 تشکیل شده‌اند.

$$Z_2^n := \{x = x_1x_2x_3\cdots x_n, |x_i = 0, 1\}. \tag{5}$$

دقت کنید که در نوشتن عناصر Z_2^n، نمادِ $x_1x_2x_3\cdots x_n$ را بجای نمادِ $(x_1, x_2, x_3, \cdots x_n)$ به کار برده‌ایم. به هر عضو از Z_2^n یک کلمه می گوییم. تعداد کلمه های Z_2^n برابر است با 2^n.

تعریف : هرگاه $e \in Z_2^n$ یک کلمه‌ی n بیتی باشد، وزن همینگ آن [1] برابر است با تعداد ۱ های آن. این وزن را با $w(e)$ نشان می دهیم. بنابراین کلمه $e = 001110$ دارای وزن 3 است، یا $w(e) = 3$.

تعریف: هرگاه $x, y \in Z_2^n$ دو کلمه‌ی n بیتی باشند، فاصله همینگ آنها برابر است با تعداد بیت هایی که با یک دیگر تفاوت دارند. به عبارت دیگر $d(x, y) := w(x - y)$. این فاصله با $d(x, y)$ نشان داده می شود و می توان نوشت:

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i| = \sum_{i=1}^{n} (x_i - y_i)^2. \tag{6}$$

براحتی می توان نشان داد که فاصله همینگ تمام خصوصیات فاصله را دارد یعنی:

$$\begin{aligned} &d(x, y) \geq 0 \ , \ \ x = y \longrightarrow d(x, y) = 0 \ , \ \ d(x, y) = 0 \longrightarrow x = y, \\ &d(x, y) = d(y, x) \\ &d(x, y) \leq d(x, z) + d(z, y). \end{aligned} \tag{7}$$

فرض کنید که کلمه ای مثل v دچار خطای e شود و تبدیل به کلمه $v' = v + e$ شود. دراین صورت تعداد خطاهای صورت گرفته دراین کلمه برابر است با تعداد 1 های درون e که برابراست با $w(e)$. دقت کنید که به ازای هر 1 در e یک خطا روی v ایجاد می شود. به عنوان مثال هرگاه $e = 00110101$ باشد به این معناست که در مکان های ۳ ، ۴ ، ٦ و ٨ کلمه v دچار خطا شده است.

هرگاه احتمال وقوع یک خطا در یک بیت برابر با p باشد و فرض کنیم که خطاهای بیت های مختلف از هم مستقل هستند آنگاه می توان احتمال وقوع یک خطای e را حساب کرد. این خطا برابراست با

$$P(e) = p^{w(e)}(1 - p)^{n - w(e)} \approx p^{w(e)}. \tag{8}$$

که در تساوی تقریبی آخر از این استفاده کرده‌ایم که p معمولا عدد بسیار کوچکی است.

[1] Hamming Weigth

هرگاه $x \in Z_2^n$ یک کلمه‌ی n بیتی باشد، کره هامینگ به شعاع d به مرکز x شامل تمام نقاطی است که فاصله آنها از x مساوی یا کمتراز d است. این کره را با $B_d(x)$ نشان می دهیم:

$$B_d(x) := \{y \in Z_2^n | d(x,y) \leq d\}. \tag{9}$$

ایده اصلی کد گذاری کانال آن است که از 2^n نقطه در فضای Z_2^n تعداد کمتری نقطه انتخاب کنیم و آنها را برای کد کردن 2^k $(k < n)$ کلمه بکار ببریم و این کار را به نحوی انجام دهیم که فاصله این کلمه ها با یک دیگر زیاد باشد به نحوی که اشتباهاتی که در طول کانال رخ می دهد آنها را تبدیل به یک دیگر نکند. بنابراین تعریف رسمی یک کد به صورت زیر است.

تعریف : یک کد عبارت است از یک زیر مجموعه از Z_2^n. این زیر مجموعه را با C نشان می دهیم و تعداد اعضای آن را برابر با 2^k می گیریم. به هر عضو C یک کد—کلمه یا $Codeword$ می گوییم. بنابراین می گوییم که مد—کلمه های k C بیت حرف را کد می کنند، چون تعدادشان برابر با 2^k است.

تعریف: فاصله یک کدِ C که آن را با $d(C)$ یا d نشان می دهیم برابر است با کمترین فاصله ای که بین کلمات آن وجود دارد. به عبارت دیگر

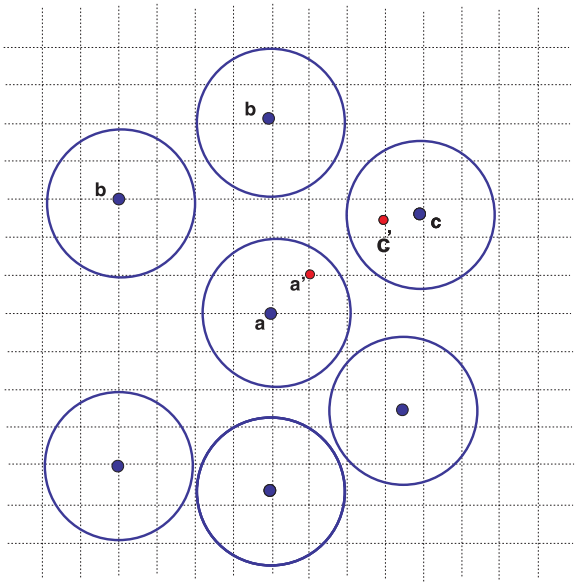
$$d(C) := min_{x,y \in C} d(x,y). \tag{10}$$

نمادگذاری: کدی را که در فضای Z_2^n نوشته می شود وبرای کد کردن k بیت به کار می رود، و فاصله آن برابر با d است با نمادِ $[n,k,d]$ نشان می دهند.

یک نحوه تصحیح خطاها در یک کد کلاسیک در شکل ۲ نشان داده شده است. هرگاه فاصله کد برابر با $d = 2t+1$ باشد، حول هر کد—کلمه یک کره هامینگ به شعاع t رسم می کنیم. دراین صورت هرگاه نقطه ای دریافت کنیم که متعلق به C نباشد به این معناست که خطایی صورت گرفته است و آن را به صورت کلمه‌ای که نزدیک ترین فاصله را با آن دارد تصحیح می کنیم. به عنوان مثال در شکل ۲ کلمه های دریافتی a' و c' متعلق به C نیستند و به کلمه های a و c تصحیح می شوند. به این ترتیب می گوییم که کدی که فاصله آن برابر با $d = 2t+1$ است قادر است که خطاهای با وزن t یا اصطلاحاً t خطا را تصحیح کند.

مثال ۱ : کد زیر را در نظر بگیرید:

$$C = \{000, 111\}. \tag{11}$$



شکل ۲: نحوه تشخیص و خنثی کردن خطا

این کد که اصطلاحاً کد تکرار سه تایی نامیده می شود کدی است از نوع $[3,1,3]$ و می تواند یک خطا را تشخیص دهد و تصحیح کند.

مثال ۲ : کد زیر را در نظر بگیرید:

$$C = \{000000, 101010, 010101, 1111111\}. \tag{12}$$

این کد از نوع $[6,2,3]$ و می تواند یک خطا را تشخیص دهد و تصحیح کند. می توان این کد را برای کد کردن دو بیت به صورت زیر به کار برد:

$$\begin{aligned} 00 &\longrightarrow 000000 \\ 01 &\longrightarrow 010101 \\ 10 &\longrightarrow 101010 \\ 11 &\longrightarrow 111111. \end{aligned} \tag{13}$$

سوال آیا می توانید کدی از نوع $[5,2,3]$ بنویسید؟

مثال ۳ : کد زیر را در نظر بگیرید:

$$C = \{000, 011, 101, 110\}. \tag{14}$$

در این کد بیت سوم که به بیت پاریته موسوم است پاریته دو بیت اول را تعیین می کند به این معنا که اگر مجموع این دو بیت برابر با صفر باشد مقدار این بیت برابر با صفر و درغیر این صورت مقدار آن برابر با ۱ است. به عبارت دیگر در هر کلمه این کد داریم $x_3 = x_1 + x_2$. با این حساب خواهیم داشت $x_1 + x_2 + x_3 = 0$، یعنی پاریته مجموع سه عدد برابر با صفر خواهد بود. به این ترتیب این کد می تواند یک خطا را آشکار کند، زیرا وقوع یک خطا پاریته کلمه را تغییر خواهد داد. اما این کد تنها می تواند این خطا را آشکار کند و قادر به تصحیح آن نیست . مثلاً معلوم نیست که کلمه دریافتی 111 کدام یک از سه کلمه ارسالی $110, 101, 011$ بوده است که دچار خطا شده است. کمی دقت نشان می دهد که فاصله این کد برابراست با 2 و به همین دلیل است که نمی تواند تشخیص دهد که یک کلمه معیوب ناشی از ایجاد خطا بر روی کدام کلمه کد بوده است زیرا کره های همینگ به شعاع یک برای سه کلمه فوق همگی نقطه‌ی 111 را در بردارند.

مثال ۴: کد زیر را در نظر بگیرید:

$$C = \{00000, 01011, 01011, 01110, 10011, 10110, 11000, 11101\}. \tag{15}$$

این کد از نوع $[5, 3, 2]$ و می تواند یک خطا را تشخیص دهد ولی نمی تواند آن را تصحیح کند. دراین کد رابطه زیر برقرار است:

$$x_4 = x_1 + x_2, \qquad x_5 = x_1 + x_2 + x_3. \tag{16}$$

بنابراین بیت چهارم پاریته دو بیت اول و بیت پنجم پاریته سه بیت اول را در خود نگاه می دارد. سه بیت اول معمولاً بیت های پیام یا $Message\ Bits$ و دو بیت آخر بیت های پاریته یا $ParityChecks$ نامیده می شوند.

$$\begin{aligned} 00 &\longrightarrow 000000 \\ 01 &\longrightarrow 010101 \\ 10 &\longrightarrow 101010 \\ 11 &\longrightarrow 111111. \end{aligned} \tag{17}$$

این کد از نوع $[6, 2, 3]$ و می تواند یک خطا را تشخیص دهد و تصحیح کند. می توان این کد را برای کد کردن دو بیت به صورت زیر به کار برد:

$$\begin{aligned} 00 &\longrightarrow 000000 \\ 01 &\longrightarrow 010101 \\ 10 &\longrightarrow 101010 \\ 11 &\longrightarrow 111111. \end{aligned} \tag{18}$$

احتمالاً اگر سعی کرده باشید که کدی از نوع [5, 2, 3] نویسید متوجه شده‌اید که نوشتن کد ها چندان آسان نیست و قاعده‌ی سرراستی برای ساختن آنها وجود ندارد. کدهای خطی [2] دسته‌ای از کد ها هستند که خواص بسیار ساده و جالبی دارند و راه و روشی سیستماتیک برای نوشتن کد ها بدست می دهند. در بخش بعدی این کد ها را معرفی می کنیم.

۱.۲ کد های خطی

می دانیم که مجموعه Z_2^n یک ساختار خطی دارد و می توان آن را به عنوان یک فضای برداری روی میدان Z_2 در نظر گرفت. هرگاه $x = x_1x_2\cdots x_n \in Z_2^n$ و $y = y_1y_2\cdots y_n \in Z_2^n$ آنگاه جمع این دو بردار به شکل زیر تعریف می شود

$$x + y = z_1z_2\cdots z_n, \tag{19}$$

که درآن

$$z_i = x_i + y_i \qquad mod\ 2.$$

هم چنین به صورت بدیهی می توان هر برداری مثل x را در عددی مثل $\alpha \in Z_2$ ضرب کرد. همه مفاهیمی که برای فضاهای برداری می شناسیم مثل استقلال خطی بردارها، پایه، بعد و نظایر آن برای این فضا نیز تعریف می شوند، با این تفاوت که بایستی همواره در نظر داشته باشیم که میدان اعداد دراینجا میدان اعداد $Z_2 = \{0, 1\}$ است. دراین فضا می توان یک ضرب داخلی نیز به شکل زیر تعریف کرد:

$$\langle x \cdot y = \sum_{i=1}^{n} x_iy_i. \tag{20}$$

تعداد عناصر این فضای خطی محدود و برابر با 2^n است.

در قسمت های قبلی کد را به صورت زیر مجموعه‌ای از Z_2^n تعریف کردیم. برای کد های خطی طبیعی است که از خاصیت خطی بودن فضای Z_2^n استفاده کنیم. به همین دلیل کد خطی را به شکل زیر تعریف می کنیم.

تعریف : در فضای خطی Z_2^n یک کد خطی چیزی نیست جز یک زیر فضای C از Z_2^n. هرگاه این زیر فضا k بعدی باشد آن را به شکل زیر برای کد کردن k بیت به کار می بریم. فرض کنید که $B = \{v_1, v_2, \cdots v_k\}$ یک پایه برای C و $\alpha = \alpha_1\alpha_2\cdots\alpha_k \in Z_2^k$ یک کلمه k بیتی باشد. دراین صورت این کلمه را به صورت زیر در n بیت کد می کنیم:

$$\alpha \longrightarrow v(\alpha) = \alpha_1v_1 + \alpha_2v_2 + \cdots\alpha_kv_k. \tag{21}$$

در چنین مواردی می نویسیم

$$C = \langle v_1, v_2, \cdots v_k\rangle. \tag{22}$$

[2] Linear codes

مثال: در فضای Z_2^5 یک کد سه بعدی در نظر بگیرید که با بردارهای پایه $B = \{00111, 11000, 10001\}$ جاروب می شود. دراین صورت کلمه‌ی سه بیتی $\alpha = \alpha_1\alpha_2\alpha_3$ به صورت زیر کد می شود:

$$\alpha \longrightarrow v(\alpha) = \alpha_1(00111) + \alpha_2(11000) + \alpha_3(10001) = (\alpha_2 + \alpha_3, \alpha_2, \alpha_1, \alpha_1, \alpha_1 + \alpha_3). \tag{23}$$

بنابراین در این کد خطی کلمات سه بیتی به صورت زیر کد می شوند:

$$\begin{aligned}
000 &\longrightarrow 00000\\
001 &\longrightarrow 10001\\
010 &\longrightarrow 11000\\
011 &\longrightarrow 01001\\
100 &\longrightarrow 00111\\
101 &\longrightarrow 10110\\
110 &\longrightarrow 11111\\
111 &\longrightarrow 01110.
\end{aligned} \tag{24}$$

نوشتن کدهای خطی بسیار آسان است، کافی است که مجموعه‌ای از بردارهای مستقل از فضای Z_2^n در نظر گرفت و کلمات را به صورت ترکیب های خطی انها کد کرد. اما معلوم نیست که انتخاب ما انتخاب خوبی باشد یعنی این کد بتواند خطاها را تصحیح کند. بنابراین سوالی که با آن مواجه هستیم آن است که چگونه می توان خواص کد های خطی نظیر فاصله را تعیین کرد و این که آیا راه ساده‌ای برای تشخیص و تصحیح خطاها توسط این کدها وجود دارد یا نه؟ در واقع بااستفاده از ساختار خطی کد می بایست بتوانیم برای این سوال ها پاسخ های روشنی بیابیم. برای پاسخ به این سوال ها می بایست ساختار کد های خطی را بهتر بشناسیم. این کاری است که در زیر بخش بعدی انجام می دهیم.

۲.۲ ساختار کد های خطی

فرض کنید که یک کد خطی با پایه $B_C = \{v_1, v_2, \cdots v_k\}$ تعریف شده باشد. دراین صورت هر کلمه‌ی α به صورت کد–کلمه‌ی $v(\alpha)$ کد می شود که درآن

$$v(\alpha) = \alpha_1 v_1 + \alpha_2 v_2 + \cdots \alpha_k v_k = (\alpha_1, \alpha_2, \cdots \alpha_k) \begin{pmatrix} v_1 \\ v_2 \\ \cdots \\ v_k \end{pmatrix}. \tag{25}$$

با تعریف ماتریس $G = \begin{pmatrix} v_1 \\ v_2 \\ \cdots \\ v_k \end{pmatrix}$ که بعدِ $k \times n$ دارد می توانیم رابطه بالا را به شکل فشرده زیر بنویسیم:

$$v(\alpha) = \alpha G. \tag{26}$$

دقت کنید که ماتریس G یک ماتریس با رتبه‌ی k است زیرا همه سطرهای آن مستقل خطی اند. ماتریس G ماتریس مولد [3] نام دارد.

مثال: کد خطی C با پایه $B_C = \{1000, 1010, 0101\}$ دارای عناصر زیر است:

$$\{0000, 1000, 1010, 0101, 0010, 1101, 1111, 0111\}. \tag{27}$$

ماتریس مولد این کد عبارت است از:

$$G = \begin{pmatrix} 1000 \\ 1010 \\ 0101 \end{pmatrix}. \tag{28}$$

می دانیم که یک کد خطی C با یک زیر فضا در فضای برداری Z_2^n مشخص می شود، شکل ۳. این کد برای نوشتن سه بیت در ۴ بیت به کار می رود. هر کلمه سه بیتی مثل $\alpha = \alpha_1\alpha_2\alpha_3$ به صورت زیر به یک کلمه چهاربیتی کد می شود:

$$\alpha \longrightarrow v(\alpha) = (\alpha_1, \alpha_2, \alpha_3) \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \alpha_1 + \alpha_2 & \alpha_3 & \alpha_2 & a_3 \end{pmatrix} \tag{29}$$

برای تشخیص خطا باید دقت کنیم که هر خطایی باعث می شود که بردارمربوط به یک کد کلمه از زیرفضای C خارج شود، شکل ۳. (دراین جا توجه به یک نکته اهمیت دارد و آن اینکه در یک کد ممکن است که یک کلمه دراثر خطا تبدیل به یک کلمه دیگر واقع در درون کد شود. هیچ کدی نمی تواند این نوع خطا ها را آشکار کرده و تصحیح کند. البته احتمال وقوع چنین خطاهایی بسیار کم است.)

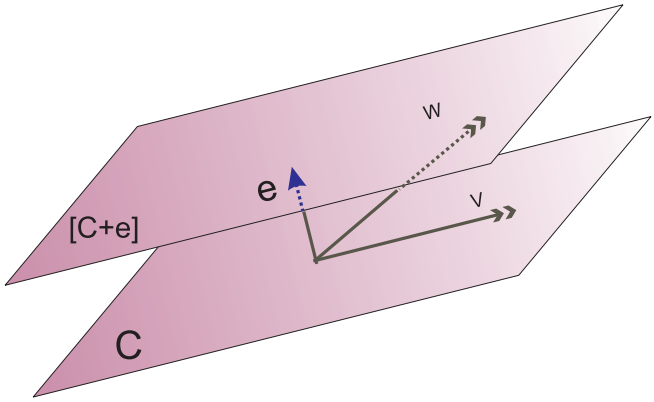
چگونه می توان تشخیص داد که برداری دچار خطا شده است؟ چگونه می توان فهمید که برداری از زیر فضای C خارج شده‌است؟ اگر به شکل ۳ نگاه کنیم یک راه ساده می توانیم برای آن پیدا کنیم. می توان از یک نشانه خیلی ساده برای این کار استفاده کرد. برای درک این نشانه احتیاج به مقدمات ساده‌ای داریم.

قضیه: هرگاه V یک فضای برداری و C یک زیر فضای آن باشد، مجموعه بردارهایی که بر C عمود هستند تشکیل یک زیرفضا می دهند.

اثبات: اثبات این قضیه بسیار ساده است .

---

[3] Generator Matrix



شکل ۳: یک خطا باعث می شود که بردار یک کلمه از زیر فضای کد خارج شود.

تعریف: اگر C یک زیرفضا باشد، فضای برداری تشکیل شده از بردارهای عمود بر C را با $C^{\perp}$ نشان می دهیم.

قضیه: اگر $C \subset V$ باشد، رابطه زیر بین ابعاد C و $C^{\perp}$ برقرار است:

$$dim(C) + dim(C^{\perp}) = n. \tag{30}$$

اثبات: فرض کنید که $dim(V) = n$ و $B_C = \{v_1, v_2, \cdots v_k\}$ یک پایه برای فضای C باشد. دراین صورت اگر $w \in C^{\perp}$ آنگاه

$$\begin{aligned} &w \cdot v_1 = 0, \\ &w \cdot v_2 = 0, \\ &\cdots \\ &w \cdot v_k = 0. \end{aligned} \tag{31}$$

این معادلات یک دستگاه از k معادله و n مجهول تشکیل می دهند که با توجه به مستقل بودن v_i ها جواب کلی آن دارای $n-k$ پارامتر آزاد خواهد بود که به این معناست که فضای جواب های این معادلات یک فضای $n-k$ بعدی است.

برای فضای $C^{\perp}$ می توان $n-k$ بردارپایه انتخاب کرد که آن ها را با $w_1, w_2, \cdots w_{n-k}$ نشان می دهیم. بنابراین خواهیم داشت

$$w_i \cdot v_j = 0, \qquad \forall\, i, j. \tag{32}$$

می توان بردارهای $w_1,\ w_2,\ w_3,\cdots w_{n-k}$ را در یک ماتریس H به صورت زیر جای داد:

$$H=\begin{pmatrix} w_1 \\ w_2 \\ \cdots \\ w_{n-k} \end{pmatrix} \longrightarrow H^T=\left(w_1^T, w_2^T, \cdots w_{n-k}^T\right) \tag{33}$$

ماتریس H یک ماتریس $times(n-k)\times n$ خواهد بود و با توجه به رابطه‌ی 32، رابطه زیر برقرار خواهد بود:

$$GH^T=0. \tag{34}$$

با توجه به این رابطه هر بردار v متعلق به کدِ C در خاصیت زیر صدق می کند:

$$vH\equiv v(\alpha)H^T=\alpha GH^T=0, \qquad \forall\alpha. \tag{35}$$

حال اگر بردار v دچار خطا شود و تبدیل به برداری مثل $w=v+e$ شود (شکل ۳)، دراین صورت خواهیم داشت

$$wH^T=(v+e)H^T=eH^T\neq 0. \tag{36}$$

بنابراین صفر نشدن wH^T به معنای آن است که بردارِ w یک بردار متعلق به کد نیست و حتماً خطایی اتفاق افتاده است. ولی چگونه می توانیم نوع خطا را بفهمیم و در جهت تصحیح آن اقدام کنیم. نکته‌ای که وجود دارد آن است که یک بردار معیوب مثل w ممکن است به طرق مختلفی تولید شده باشد، بعنوان مثال این بردار حاصل خطای e_1 بر بردار v_1 یا خطای e_2 بر بردار v_2 باشد به نحوی که

$$v_1+e_1=v_2+e_2=w. \tag{37}$$

از میان خطاهای ممکن می بایست محتمل ترین خطا یعنی خطای با کمترین وزن هامینگ در نظر گرفت و فرض کرد که $w=v+e_0$ که در آن e_0 کمترین وزن ممکن را دارد و درنتیجه می بایست w را به شکل زیر تصحیح کرد:

$$w\longrightarrow w+e_0. \tag{38}$$

می توان مطالب بالا را به زبان دقیق و ریاضی نیز بیان کرد. اگر به شکل ۳ نگاه کنیم متوجه می شویم که C یعنی خود کُد، یک زیر فضاست و بقیه صفحات در واقع یکسان با C هستند ولی زیر فضا نیستند زیرا شامل بردار 0 نیستند. برای پیشتر رفتن به تعاریف زیر احتیاج داریم.

تعریف: فرض کنید که V یک فضای برداری و C یک زیر فضای آن باشد. دراین صورت هردو بردار $w_1,w_2\in V$ را هم ارز می گوییم هرگاه رابطه زیر برقرارباشد:

$$w_1-w_2\in C. \tag{39}$$

خواننده براحتی می تواند ثابت کند که این رابطه یک رابطه هم ارزی است و بنابراین فضای V (دراینجا Z_2^n) توسط زیرفضای C (که دراینجا همان فضای کد—کلمه‌هاست) به زیرمجموعه هایی که عناصر آنها بایکدیگر هم ارز هستند افراز می شود. هر

کلاس هم ارز یک $Coset$ نامیده می شود. اگر به شکل ۳ نگاه کنیم متوجه می شویم که از نظر هندسی هر هم مجموعه یک صفحه موازی با صفحه‌ی C است. باید دقت کنیم که شکل ۳ تنها تا حدودی به درک شهودی ما کمک می کند و از بعضی جهات می تواند گمراه کننده باشد، زیرا ما دراینجا با یک فضای برداری روی میدان $Z_2 = \{0, 1\}$ سروکار داریم نه یک فضای برداری حقیقی. به همین دلیل تعداد بردارهای کل فضا و هم چنین زیرفضای C و تمام کلاس های هم ارزی محدود است. فرض کنید که بردارهای فضای کد یعنی C را به صورت زیر نمایش دهیم:

$$C = \{v_1, v_2, v_3, \cdots v_K\} \tag{40}$$

که درآن $K = 2^k$ (زیرا فرض کرده ایم که C یک فضای k بعدی است.). دراین صورت به ازای هر برداری مثل e_i که متعلق به C نباشد یک کلاس هم ارزی داریم که برابر است با:

$$[e_i] \equiv e_i + C = \{v_1 + e_i, v_2 + e_i, v_3 + e_i, \cdots v_K + e_i\}. \tag{41}$$

به دو نکته باید دقت کرد:

نکته اول : برای تمام عناصر یک کلاس اثر H یکسان است، و برابر است با e_iH . هم چنین اثر H روی هیچ دو کلاس متفاوتی یکسان نیست، زیرا

$$w_1H^T = 0, \quad w_2H^T = 0, \ \longrightarrow \ (w_1 - w_2)H^T = 0, \ \longrightarrow \ w_1 - w_2 \in C, \ \longrightarrow [w_1] = [w_2]. \tag{42}$$

بنابراین اثر H روی یک کد دریافتی تنها به کلاس هم ارزی بستگی دارد و می توان آن را به عنوان شناسنده کلاس هم ارزی یا نشانه خطا[4] بکار برد.

نکته دوم : یک کلاس هم ارزی مثل کلاس 41 را می توان به صورت کلاس $[e_i + v_1]$ یا کلاس $[e_i + v_2]$ یا کلاس $[e_i + v_K]$ نیز نامید و در وهله اول هیچ ترجیحی در نامگذاری یک کلاس به یکی از این صورت ها نیست. از این به بعد نماینده کلاس را برداری می گیریم که کمترین وزن هامینگ را دارد. بنابراین وقتی می گوییم کلاس $[e_i]$ یعنی بردار e_i دراین کلاس کمترین وزن هامینگ را دارد.

دو نکته فوق به ما می آموزند که چگونه باید خطاها را آشکار کرده و تصحیح کنیم.

نحوه تصحیح خطا: هر بردار w که دریافت می شود، wH^T برای آن محاسبه می شود. اگر $wH^T = 0$ باشد به این معناست که خطایی صورت نگرفته است. اگر $wH^T \neq 0$ می فهمیم که خطایی صورت گرفته است. مقدار wH در واقع نشانه خطاست و برای ما کلاس هم ارزی $[e_i]$ را تعیین می کند. دراین کلاس e_i کمترین وزن هامینگ را دارد و بنابراین محتمل ترین خطا همان e_i است . بنابراین کلمه دریافتی w را به صورت $v = w+e_i$ تصحیح می کنیم.

[4] Error Syndrome

۳.۲ خواص بیشتری از کد های خطی

مرور کوتاه خود از کد های خطی را با بیان چند خاصیت مهم از آنها به پایان می بریم.

۱.۳.۲ کدهای خود—دوگان

از آنجا که فضای برداری Z_2^n یک فضای برداری حقیقی نیست و روی میدان $Z_2\{0,1\}$ ساخته شده‌است، خواص آن می تواند با خواص فضاهای برداری حقیقی تمایز اساسی داشته باشد. به عنوان مثل در Z_2^n یک بردار می تواند بر خود عمود باشد مثل بردار $\{0011\} \in Z_2^4$. دلیل این امر این است که در میدان اعداد حقیقی مجموع چند عدد مثبت نمی تواند صفر شود ولی در میدان Z_2 چنین چیزی ممکن است. قبلاً گفتیم که اگر $C \subset V$ یک فضای برداری باشد، $C^\perp$ یعنی مجموعه تمام بردارهای عمود بر C یک زیر فضا از V است. تصور هندسی ای که ما در مورد فضای $C^\perp$ داریم از فضاهای حقیقی ریشه گرفته‌است. در مورد فضای Z_2^n وضعیت کاملاً متفاوتی حاکم است. به عنوان مثال C می تواند زیر فضایی از $C^\perp$ شود و یا با آن مساوی شود.

مثال: در Z_2^4 قرار می دهیم:

$$C = \langle 0011, 1100 \rangle = \{0000, 0011, 1100, 1111\} \tag{43}$$

به آسانی معلوم می شود که

$$C^\perp = \{0000, 0011, 1100, 1111\} = C. \tag{44}$$

چنین کدی را یک کدِ خود—دوگان می نامیم.

مثال: در Z_2^4 قرار می دهیم:

$$C = \langle 0011 \rangle = \{0000, 0011\}, \tag{45}$$

دراین صورت داریم

$$C^\perp = \{0000, 0011, 1100, 1111\} \supset C. \tag{46}$$

۲.۳.۲ فرم استاندارد ماتریس های مولد و پاریته

دیدیم که در یک کد خطی C ماتریس مولد از بردارهای پایه ساخته می شود. هرگاه $B_C = \{v_1, v_2, \cdots v_k\}$ یک پایه برای C باشد، ماتریس مولدِ این کد که آن را با G نمایش می دهیم به صورت زیر نوشته می شود:

$$G = \begin{pmatrix} v_1 \\ v_2 \\ \cdots \\ v_k \end{pmatrix}. \tag{47}$$

از آنجا که بردارهای پایه از هم مستقل هستند نتیجه می گیریم که رتبه ماتریس G، یعنی تعداد سطرهای غیر صفر آن، برابر با k است. هم چنین می توانیم بجای پایه فوق پایه دیگری مثل $\{u_1, u_2, \cdots u_k\}$ به کار ببریم که بردارهای آن با عوض کردن ترتیب بردارهای پایه اولیه و یا ترکیبی خطی از آنها بدست آمده است. این امر به این معناست که می توان با انتخاب پایه مناسب ماتریس G را به شکل استانداردِ زیر درآورد:

$$G = \begin{pmatrix} I_k & | & X_{k,n-k} \end{pmatrix}, \tag{48}$$

که درآن I_k ماتریس یکه با بعد k است و $X_{k,n-k}$ یک ماتریس با بعد $k \times n-k$ است. تحت این شرایط معلوم می شود که ماتریس H به شکل زیر در می آید زیرا می بایست در شرط $GH^T = 0$ صدق کند:

$$H^T = \begin{pmatrix} X_{k,n-k} \\ -- \\ I_{n-k} \end{pmatrix} \longrightarrow H = \begin{pmatrix} X^T_{n-k,k} & | & I_{n-k} \end{pmatrix}. \tag{49}$$

از رابطه‌ی $GH^T = 0$ نتیجه می گیریم که $HG^T = 0$. این رابطه بیان می کند که یک کد دیگر وجود دارد که ماتریس مولد آن H و وماتریس پاریته آن G است. اما این کد چیزی نیست جز $C^\perp$.

۳.۳.۲ یک رابطه تعامد مهم بین یک کد و کد عمود بر آن

دراین زیربخش یک رابطه مهم بین دو کد C و $C^\perp$ را ثابت می کنیم که بعدها درساختن کد های کوانتومی نقش مهمی ایفا خواهد کرد. نخست به اتحاد زیر توجه می کنیم:

$$\sum_{x \in \{0,1\}} (-1)^{xy} = 2\delta_{y,0} \tag{50}$$

و یا تعمیم آن به n بیت که به صورت زیر بیان می شود:

$$\sum_{x \in \{0,1\}^n} (-1)^{x \cdot y} = 2^n \delta_{y,0}. \tag{51}$$

دقت کنید که هرگاه به $\mathbf{x}$ و $\mathbf{y}$ به صورت بردارهای سطری فکر کنیم آنگاه می توانیم بنویسیم $\mathbf{x} \cdot \mathbf{y} = \mathbf{x}\mathbf{y}^T$. این نکته در درک رابطه بعدی اهمیت دارد.

می خواهیم ببینیم که اگر در رابطه بالا بجای همه $\mathbf{x} \in Z_2^n$ تنها روی کد کلمه های درون C که زیرفضایی از Z_2^n است، جمع بزنیم چه چیزی بدست می آوریم.
دقت می کنیم که به ازای هر $\mathbf{y} \in Z_2^n$

$$\sum_{v(\alpha)\in C} (-1)^{v(\alpha)\cdot y} = \sum_{\alpha\in\{0,1\}^k} (-1)^{\alpha G\cdot y}$$
$$= \sum_{\alpha\in\{0,1\}^k} (-1)^{\alpha G y^T} = 2^k \delta_{Gy^T,0} = 2^k \delta_{yG^T,0}. \quad (52)$$

اما می دانیم که G ماتریس پاریته کد $C^\perp$ است و درنتیجه هرگاه $\mathbf{y}G^T$ برابر با صفر باشد به این معناست که $\mathbf{y} \in C^\perp$. بنابراین رابطه بالا را می توان به صورت زیر نوشت:

$$\sum_{v(\alpha)\in C} (-1)^{v(\alpha)\cdot y} = \begin{cases} 2^k & if \;\; y \in C^\perp \\ 0 & if \;\; y \notin C^\perp \end{cases} \quad (53)$$

از این رابطه در بخش کد های کوانتومی استفاده خواهیم کرد.

۴.۳.۲ د های خطی فاصله در ک

دیدیم که فاصله‌ی یک کُد نقش مهمی در خواص آن کد برای آشکارسازی و اصلاح خطاها دارد. به عبارت دقیق تر دیدیم که یک کُد با فاصله‌ی $d = 2t + 1$ می تواند خطاهای با وزن t و کمتر از آن را اصلاح کند. سوال طبیعی‌ای که در اینجا مطرح است آن است که چگونه می توان فاصله‌ی یک کُد خطی را با استفاده از ماتریس های مولد یا پاریته‌ی آن بدست آورد. پاسخ این سوال در قضیه زیر داده شده‌است.

قضیه: در یک کد خطی با ماتریس پاریته‌ی H، فاصله کد برابر است با d اگر هر $d-1$ تا از ستون های H از هم مستقل خطی باشند و d ستون وابسته خطی در آن وجود داشته باشد.

اثبات: بنابرتعریف می دانیم که فاصله یک کُد کمترین مقدار فاصله همینگ بین کلمات آن است. در یک کد خطی فاصله برابر می شود با کمترین وزن همینگ برای بردارها. فرض کنید که $v \in C$ برداری با وزن e باشد. بنابراین بردار v دارای e مولفه با مقدارِ 1 است. این مولفه ها را با $v_{i_1}, v_{i_2}, \cdots v_{i_e}$ نشان می دهیم. ستون های ماتریس H را با $c_1, c_2, \cdots c_n$ نشان می دهیم. یعنی

$$H = \begin{pmatrix} c_1 & c_2 & \cdots & \cdots & c_n \end{pmatrix}. \quad (54)$$

می دانیم که برای هر کلمه متعلق به کُد C شرط $vH^T = 0$ برقراراست. بنابراین داریم

$$v_{i_1} c_{i_1}^T + v_{i_2} c_{i_2}^T + \cdots v_{i_e} c_{i_e}^T = 0. \quad (55)$$

اما این امر به این معناست که بردارهای c_{i_1} ، c_{i_2} تا c_{i_n} وابسته خطی هستند. بنابراین به ازای هر بردار با طول همینگ e در ماتریس H، e تاستون وجود دارد که باهم وابسته خطی هستند. پس اگر هر $d-1$ ستونی از H مستقل خطی باشند، نتیجه می گیریم که فاصله کد بزرگتر یا مساوی با d است. حال اگر d تا ستون پیدا کنیم که وابسته خطی باشند به این معناست که یک بردار با وزن همینگ d در کُد وجود دارد و بنابراین فاصله کد برابراست با d .

مثال: کُدی درنظر بگیرید با ماتریس پاریته‌ی زیر:

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \tag{56}$$

براحتی معلوم می شود که در این ماتریس هر دو ستونی مستقل خطی اند. علاوه بر این می توان سه ستون یافت که وابسته خطی باشند مثل ستون های ۳، ۴ و ۷. بنابراین فاصله این کد برابر است با ۳.

۳ کدهای کوانتومی

پس از این مقدمه در باره کد های کلاسیک می توانیم کد های کوانتومی را مطالعه کنیم. همانطور که در مقدمه گفتیم خطاهای کوانتومی و نحوه اصلاح آنها بکلی متفاوت با خطاهای کلاسیک هستند و به همین جهت در ابتدای امر گمان می شد که کامپیوترهای کوانتومی هرگز ساخته نخواهند شد زیرا هیچ روشی برای شناسایی و اصلاح خطاهای ناگزیری که درآنها وجود دارد شناخته شده نبود. امروزه معلوم شده است که یک نظریه جامع و کامل برای آشکارسازی و اصلاح خطاهای کلاسیک وجود دارد که ساخت عملی کامپیوترهای کوانتومی را امکان پذیر می کند. اندیشه هایی که مبنای این نظریه قرار گرفته اند اساساً از همان نظریه کلاسیک الهام گرفته شده‌اند ولی در تدوین آنها تمام محدودیت های مکانیک کوانتومی نظیر عدم امکان اندازه گیری های دلخواه و یا عدم امکان تکثیر حالت ها مورد ملاحظه قرار گرفته‌اند. کاری که ما دراین بخش می کنیم آن است که نخست کدهای ساده‌ای را معرفی می کنیم که قادرند خطاهای معینی را آشکار و اصلاح کنند. سپس به تدریج که این کد ها را بسط می دهیم اصول اساسی نظریه اصلاح خطاهای کوانتومی را نیز دریک چارچوب کلی شرح خواهیم داد.

۱.۳ اصلاح خطاهای بیت — برگردان

فرض کنید که تنها امکانی که برای خطا وجود دارد برگردان یک کیوبیت است به این معنا که در طول کانال ممکن است عملگر X (یاهمان σ_x) روی یک کیوبیت اثر کند. دراین صورت هرگاه یک کیوبیت در ابتدای کانال به صورت $|\psi\rangle = a|0\rangle + b|1\rangle$ ارسال شده باشد، در اثر خطای احتمالی ممکن است به صورت $X|\psi\rangle = a|1\rangle + b|0\rangle$ دریافت شود. برای آشکارسازی این نوع خطا کیوبیت های $|0\rangle$ و $|1\rangle$ را به صورت زیر کُد می کنیم:

$$|0\rangle \longrightarrow |0\rangle_E = |000\rangle, \qquad |1\rangle \longrightarrow |1\rangle_E = |111\rangle, \tag{57}$$

که در آن علامت E از کلمه $Encoding$ گرفته شده است. دقت کنید که این امر نقض کننده قضیه عدم تکثیر حالت نیست. درنتیجه حالت یک کیوبیت به صورت زیر کد می شود:

$$|\psi\rangle = a|0\rangle + b|1\rangle \longrightarrow |\psi\rangle_E = a|000\rangle + b|111\rangle. \tag{58}$$

این حالت ویژه حالت مشترک عملگرهای Z_1Z_2 و Z_1Z_3 با مقادیر ویژه ۱ است. به عبارت دیگر

$$Z_1Z_2|\psi\rangle_E = |\psi\rangle_E, \qquad Z_1Z_3|\psi\rangle_E = |\psi\rangle_E. \tag{59}$$

حال اگر یک خطای X_1 ، X_2 یا X_3 رخ دهد حالت $|\psi\rangle_E$ تبدیل به حالتی می شود که ویژه مقدارش برای عملگرهای فوق تغییر خواهد کرد.

$$\begin{aligned}
I|\psi\rangle_E = a|000\rangle + b|111\rangle \quad &\longrightarrow \quad Z_1Z_2 = 1, \;\; Z_1Z_3 = 1 \\
X_1|\psi\rangle_E = a|100\rangle + b|011\rangle \quad &\longrightarrow \quad Z_1Z_2 = -1, \;\; Z_1Z_3 = -1 \\
X_2|\psi\rangle_E = a|010\rangle + b|101\rangle \quad &\longrightarrow \quad Z_1Z_2 = -1, \;\; Z_1Z_3 = 1 \\
X_3|\psi\rangle_E = a|001\rangle + b|110\rangle \quad &\longrightarrow \quad Z_1Z_2 = 1, \;\; Z_1Z_3 = -1.
\end{aligned} \tag{60}$$

در روابط بالا مقادیر ویژه عملگرهای Z_1Z_2 و Z_1Z_3 را نوشته ایم. این عملگرها که مقادیر ویژه آنها خطا را آشکار می کنند نشانه های خط ا [5] نامیده می شوند. نکته مهم آن است که حالت دچار خطاشده همچنان ویژه بردار عملگرهای Z_1Z_2 و Z_1Z_3 باقی می ماند بنابراین اندازه گیری این دو عملگر باعث فروکاهش تابع موج نمی شود. بعد از اینکه معلوم شد خطای بیت برگردان در چه نقطه ای صورت گرفته است می توان با اعمال عملگر تصحیح کننده خطا را اصلاح کرد. دراین جا با چند سوال مواجه می شویم.

۱: چگونه می توان با یک مدار کوانتومی کدِ 58 را ایجاد کرد؟

۲: چگونه می توان عملگرهای نشانه‌ی خطا را اندازه گرفت؟ چگونه می توان این کار را به صورت منظمی انجام داد؟

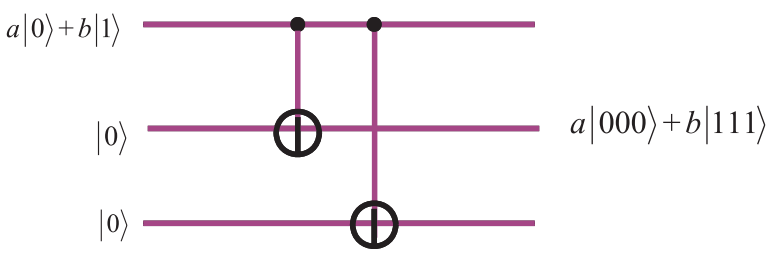
پاسخ سوال اول بسیار ساده است. این کار را می توان با استفاده از عملگر $CNOT$ انجام داد. از این به بعد برای یک عملگر $CNOT$ که بیت کنترلی اش i و بیت هدف اش j است، نمادِ C_{ij} را به کار می بریم. درنتیجه خواهیم داشت:

$$C_{12}C_{13}(a|0\rangle + b|1\rangle)|00\rangle = a|000\rangle + b|111\rangle. \tag{61}$$

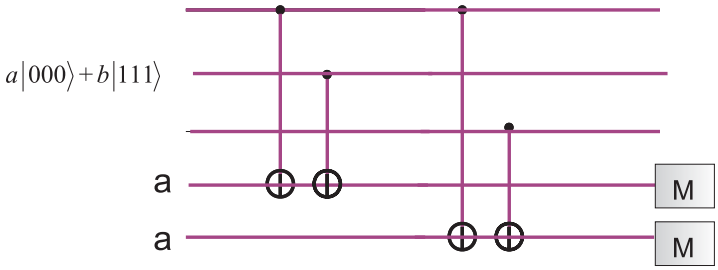
مدار شکل ۴ نشان می دهد که این کار چگونه انجام می شود.

برای سوال دوم نیز بهترین کار آن است که اطلاعات مربوط به نشانه های خطا در کیوبیت های جداگانه‌ای نوشته شود. این کیوبیت ها اصطلاحاً کیوبیت های خدمتکار [6] نامیده می شوند. برای این کار باز هم از عملگر $CNOT$ استفاده می کنیم. فرض

[5] Error Syndrome
[6] Ancilla qubit



شکل ۴: مداری که کد اصلاح کننده بیت – برگردان را درست می کند.



شکل ۵: مداری که نشانه های خطای بیت – برگردان را تشخیص می دهد.

کنید که یک بیت خدمتکار را در حالت $|0\rangle$ نگاه داشته باشیم. می خواهیم کاری کنیم که اگر در بیت های اول و دوم کد یک بیت برگردانده شده باشد، مقدار این بیت خدمتکار به یک تغییر کند. کافی است که به رابطه زیر توجه کنیم:

$$C_{1a}C_{2a}|ijk\rangle|0\rangle_a = |ijk\rangle|i+j\rangle_a. \tag{62}$$

دراین جا نماد a برای بیت خدمتکار بکار می رود. بنابراین اگر i و j باهم مساوی باشند، مقدار بیت خدمتکار برابر با صفر و درغیر این صورت برابر با یک خواهد بود. مدار شکل ۵ که در مقصد عمل می کند، نشان می دهد که این کار چگونه انجام می شود.

۴ مثال هایی از کدهای کوانتومی

کد تصحیح کننده خطای $bit - flip$
کد تصحیح کننده خطای $phase - flip$
کد تصحیح کننده همه خطاها

---

۵ نظریه تصحیح/خطا

$$(\mathcal{R} \circ \mathcal{E})() \propto \rho \tag{63}$$

قضیه: فرض کنید که C یک کد کوانتومی و P تصویرگر روی C باشد. هم چنین فرض کنید که $\mathcal{E}$ یک فرایند کوانتومی با عناصرِ E_i باشد. دراین صورت شرط لازم و کافی برای آنکه یک فرایند تصحیح خطای $\mathcal{R}$ وجود داشته باشد که خطای $\mathcal{E}$ را روی C تصحیح کند آن است که

$$PE_i^\dagger E_j P = \alpha_{ij} P, \tag{64}$$

که درآن α با درایه های α_{ij} یک ماتریس هرمیتی است.
عناصر E_i خطاهای کوانتومی خوانده می شود. درصورتیکه فرایند $\mathcal{R}$ وجود داشته باشد، این خطا ها خطاهای تصحیح پذیر خوانده می شود.

$$PF_k^\dagger F_l P = \sum_{ij} u_{ki}^\dagger u_{jl} PE_i^\dagger E_j P. \tag{65}$$

$$PF_k^\dagger F_l P = d_{kl} P \tag{66}$$

$$P_l P_k = P_l^\dagger P_k = \frac{U_l P F_l^\dagger F_k P U_k^\dagger}{\sqrt{d_{ll} d_{kk}}} = 0 \tag{67}$$

$$\begin{aligned} U_k^\dagger P_k F_l \sqrt{\rho} &= U_k^\dagger P_k^\dagger F_l P \sqrt{\rho} = \frac{U_k^\dagger U_k P F_k^\dagger F_l P \sqrt{\rho}}{\sqrt{d_{kk}}} \\ &= \delta_{kl} \sqrt{d_{kk}} P \sqrt{\rho} \\ &= \delta_{kl} \sqrt{d_{kk}} \sqrt{\rho}. \end{aligned} \tag{68}$$

$$\begin{aligned} \mathcal{R}(\mathcal{E}(\rho)) &= \sum_{kl} U_k^\dagger P_k^\dagger F_l \rho F_l^\dagger P_k U_k \\ &= \sum_{kl} \delta_{kl} d_{kk} \rho \propto \rho. \end{aligned} \tag{69}$$

$$\mathcal{R}(\mathcal{E}_C(\rho)) \propto P \rho P. \tag{70}$$

$$\sum_{ij} R_j E_i P \rho P E_i^\dagger R_j^\dagger = c P \rho P \tag{71}$$

$$R_k E_i P = c_{ki} P \tag{72}$$

$$P E_i^\dagger E_j P = \alpha_{ij} P \tag{73}$$

$$|x + C_2\rangle := \frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} |x + y\rangle \tag{74}$$

$$\frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} (-1)^{(x+y)\cdot e_2} |x + y + e_2\rangle \tag{75}$$

$$\frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} (-1)^{(x+y)\cdot e_2} |x + y + e_1\rangle |H_1 e_1\rangle \tag{76}$$

$$\frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} (-1)^{(x+y)\cdot e_2} |x + y + e_1\rangle \tag{77}$$

$$\frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} (-1)^{(x+y)\cdot e_2} |x + y\rangle \tag{78}$$

$$\frac{1}{\sqrt{|C_2|2^n}} \sum_{z} \sum_{y \in C_2} (-1)^{(x+y)\cdot (e_2+z)} |z\rangle \tag{79}$$

$$\frac{1}{\sqrt{|C_2|2^n}} \sum_{z'} \sum_{y \in C_2} (-1)^{(x+y)\cdot z'} |z' + e_2\rangle \tag{80}$$

$$\frac{1}{\sqrt{2^n/|C_2|}} \sum_{z' \in C_2^{\perp}} (-1)^{x \cdot z'} |z' + e_2\rangle \tag{81}$$

$$\frac{1}{\sqrt{2^n/|C_2|}} \sum_{z' \in C_2^{\perp}} (-1)^{x \cdot z'} |z'\rangle \tag{82}$$

$$\frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} |x + y\rangle \tag{83}$$

۷ کد های پایدار کننده

۱.۷ صورت بندی پایدارکننده ها

۲.۷ گیت های یکانی در صورت بندی پایدارکننده ها

۳.۷ اندازه گیری در صورت بندی پایدارکننده ها

۴.۷ قضیه گوتزمان — نیل

۵.۷ ساختن کد های پایدارکننده

٦.۷ مثال ها

۷.۷ مدارهای کوانتومی برای کد گزاری ، کد گشایی و تصحیر

۸ مصاحبه مصون از خطا

۱.۸ مصونیت از خطا — تصویر کلی

۲.۸ منطق کوانتومی مصون از خطا

۳.۸ اندازه گیری مصون از خطا