হামিং কোড (Hamming code) ঃ এরর সংশোধনের জন্য ব্যবহৃত কোডের মধ্যে বহুল ব্যবহৃত কোড হলো হামিং কোড। এতে n সংখ্যক ডাটার মধ্যে k সংখ্যক প্যারিটি বিট যুক্ত করে n + k সংখ্যক বিটের ডাটা তৈরি করা হয়। এ বিটের পজিশন/অবস্থানগুলো 1 থেকে n + k এর মধ্যে হয়। এর মধ্যে 2 এর ঘাত আকারে যে মান পাওয়া যায়, তা দ্বারা প্যারিটি বিটের পজিশন নির্ধারণ করা হয়, অর্থাৎ 1, 2, 4, 8 ইত্যাদি। আর বাকি পজিশন ডাটার বিটগুলো থাকে। যে-কোনো মানের/লেংথের ডাটাতে এ কোড ব্যবহৃত হয়। হ্যামিং কোডে প্যারিটি বিটের সংখ্যা নির্ণয়ের জন্য ব্যবহৃত সূত্র হলো– $2^k \geq n + k + 1$।

উদাহরণ ঃ ধরি, 7 (সাত) বিটের একটি ডাটা 1011001, এতে 4টি প্যারিটি বিট যুক্ত করা হয়েছে যার পজিশন যথাক্রমে ১ম, ২য়, ৪র্থ ও ৮ম। ফলে ডাটাটি হয় নিম্নরূপ–

1	0	1	P_4	1	0	0	P_3	1	P_2	P_1
11	10	9	8	7	6	5	4	3	2	1

→বিট পজিশন

হামিং বিটগুলো নির্ণয় করার জন্য কোডের যে-সব স্থানে 1 আছে, তা শনাক্ত করা হয়। এক্ষেত্রে 3, 7, 9 এবং 11 নং স্থানে রয়েছে। এখন এ সংখ্যাগুলোকে বাইনারিতে রূপান্তর করে যোগ করলে নির্ণেয় হামিং কোডগুলো পাওয়া যাবে। তবে যোগ করার সময় হাতের অঙ্ককে বাদ দিতে হবে।

এখানে,
3 = 0 0 1 1
7 = 0 1 1 1
9 = 1 0 0 1
11 = 1 0 1 1

0 1 1 0
↓ ↓ ↓ ↓
P_4 P_3 P_2 P_1

অর্থাৎ $P_1 = 0, P_2 = 1, P_3 = 1$ এবং $P_4 = 0$

এ বিটগুলো P এর স্থলে বসিয়ে হামিং কোড পাওয়া যাবে। সুতরাং নির্ণেয় হামিং কোড–

1	0	1	0	1	0	0	1	1	1	0
11	10	9	8	7	6	5	4	3	2	1

→বিট পজিশন

রিসিভারে এ কোডের যে সব স্থানে 1 আছে যে সব স্থানের সংখ্যা মানকে বাইনারিতে রূপান্তর করে তা আবার যোগ করে পরীক্ষা করে দেখতে হবে যে, যোগফল 0 কি না। যদি যোগফল 0 হয়, তবে বুঝা যায় যে, গৃহীত ডাটাতে কোন ভুল নেই, নতুবা ভুল আছে। উক্ত হামিং কোডের 2, 3, 4, 7, 9 এবং 11নং স্থানে 1 রয়েছে।

৩.৫ ভুল নির্ণয় ও সংশোধনের ক্ষেত্রে হামিং কোডের ভূমিকা (Describe the error detection and correction procedure with humming code) ঃ

ভুল নির্ণয় ও সংশোধনের ক্ষেত্রে হামিং কোডের ভূমিকা (The error detection and correction with hamming code) ঃ

বাইনারি বা অন্যান্য কোড ব্যবহার করে যোগাযোগ প্রকৌশলে তথ্য প্রেরণ করা হয়ে থাকে। নানা কারণে কোড পরিবর্তিত হয়ে গ্রাহক যন্ত্রে ভুল আকারেই ধরা পড়ে। এই ত্রুটি সংশোধনের জন্য নানা প্রকার কোড ব্যবহার করা হয়। হামিং কোড (Hamming code) এগুলোর মধ্যে উল্লেখযোগ্য। এই কোডে ডাটার মধ্যে পূর্বনির্ধারিত স্থানে একাধিক প্যারিটি বিট ব্যবহার করা হয়। হামিং প্যারিটি বিটের অবস্থান 2 দ্বারা নির্ধারণ করা হয়। n-এর মান 0, 1, 2, 3, 4 ইত্যাদি বসিয়ে 1, 2, 4, 8, 16 ইত্যাদি স্থানে কোডের মধ্যে প্যারিটি বিট ঢুকানো হয়। একটি উদাহরণের সাহায্যে হামিং কোড গঠনের পদ্ধতি ব্যাখ্যা করা হবে। Y এর ASCII কোড 1011001 এর ভিতর ১ম, ২য়, ৪র্থ এবং ৮ম স্থানে চারটি হামিং প্যারিটি বিট স্থাপন করা যায়। যেমন–

1	0	1	h_4	1	0	0	h_3	1	h_2	h_1
11	10	9	8	7	6	5	4	3	2	1

হামিং বিটগুলো নির্ণয় করার জন্য কোডের যে যে স্থানে 1 আছে, তা শনাক্ত করা হয়। এক্ষেত্রে 3, 7, 9 এবং 11 নং স্থানে রয়েছে। এখন এই সংখ্যাগুলোকে বাইনারিতে রূপান্তর করে যোগ করলে নির্ণেয় হামিং কোডগুলো পাওয়া যাবে। যোগ করার সময় ক্যারি অগ্রাহ্য করতে হবে।

3 = 0 0 1 1
7 = 0 1 1 1
9 = 1 0 0 1
11 = 1 0 1 1

0 1 1 0
h_4 h_3 h_2 h_1

এ বিটগুলো h এর স্থানে বসিয়ে হ্যামিং কোড পাওয়া যাবে। সুতরাং নির্ণেয় হ্যামিং কোড,

1	0	1	0	1	0	0	1	1	1	0
11	10	9	8	7	6	5	4	3	2	1

গ্রাহক যন্ত্র এ কোডের যে যে স্থানে 1 আছে, সে স্থানের সংখ্যা মানকে বাইনারিতে প্রকাশ করে যোগ করে পরীক্ষা করে দেখে, যোগফল 0 কি না। যদি যোগফল 0 হয়, তবে বুঝা যাবে যে, গৃহীত তথ্যের কোন ত্রুটি নেই। আলোচ্য হ্যামিং কোডটির 2, 3, 4, 7, 9 এবং 11 নং স্থানে 1 রয়েছে। কারি অ্যাহ্য করে এই সংখ্যাগুলোর বাইনারি যোগ করলে যোগফল 0 হবে। যেমন–

2	=	0	0	1	0
3	=	0	0	1	1
4	=	0	1	0	0
7	=	0	1	1	1
9	=	1	0	0	1
11	=	1	0	1	1
		0	0	0	0

যদি কোন কারণে গৃহীত কোড ত্রুটিমুক্ত হয়, তবে 1 এর স্থানগুলোর বাইনারি যোগফল 0 হবে না। মনে করি, ত্রুটির কারণে 5 নং স্থানে 1 হয়ে গেছে; গ্রাহক যন্ত্রে 1 এর স্থানগুলোর বাইনারি যোগফল হবে–

2	=	0	0	1	0
3	=	0	0	1	1
4	=	0	1	0	0
5	=	0	1	0	1
7	=	0	1	1	1
9	=	1	0	0	1
11	=	1	0	1	1
		0	1	0	1

যোগফল 5 এবং এটি নির্দেশ করে যে 5 নং স্থানে গৃহীত বিট সঠিক নয়। উক্ত বিটটি উল্টিয়ে দিলে গৃহীত কোড সংশোধিত হবে।

এ প্যারিটি ব্যবহার করে মূলত কোন তথ্যের ভুল নির্ণয় করা যায়, অর্থাৎ ভুল আছে কি না তা জানা যায়। কিন্তু কোন বিটে সেই ভুল আছে তা বের করা বা তা সংশোধন করা যায় না। এটি সংশোধনের জন্য বিভিন্ন পদ্ধতি ব্যবহার করা হয়। কোডিং পদ্ধতি এর জন্য ব্যবহার করা হয়। যেমন– হ্যামিং কোড।

অনুশীলনী-৩

অতি সংক্ষিপ্ত প্রশ্নোত্তর ঃ

১। **গ্রে-কোড বলতে কী বুঝায়?**
অথবা, গ্রে-কোড কী?
অথবা, Gray code-এর অর্থ কী? [বাকাশিবো-২০১০(পরি), ১২, ১৬]
উত্তর: এটি এমন এক ধরনের কোড, যাতে একটি সংখ্যা হতে পরবর্তী সংখ্যায় যেতে মাত্র একটি বিটের পরিবর্তন ঘটে।

২। **প্যারিটি বিট কাকে বলে?** [বাকাশিবো-২০০২, ০৩, ০৪, ০৯, ১০, ১১]
অথবা, প্যারিটি বলতে কী বুঝ? [বাকাশিবো-২০১২]
অথবা, Parity bit কী? [বাকাশিবো-২০১৪(পরি)]
অথবা, প্যারিটি বিট কী? [বাকাশিবো-২০১০(পরি), ১৫]
উত্তর: প্রেরিত তথ্যের বাইনারি কোডের ত্রুটি শনাক্তকরণের সহজতম উপায়ই হলো প্যারিটি বিট এর ব্যবহার। প্যারিটি বিট হলো একটি অতিরিক্ত বিট, যা একটি কোড গ্রুপের সাথে এক স্থান হতে অন্য স্থানে পাঠানো হয়।

৩। **Gray Code এর ব্যবহার লেখ।** [বাকাশিবো-২০১০]
উত্তর: এটি কম্পিউটারের ইনপুট ও আউটপুট ডিভাইস এবং অ্যানালগ টু ডিজিটাল কনভার্টারে ব্যবহৃত হয়।

৪। **Gray Code ব্যবহারের সুবিধা কী?**
অথবা, গ্রে-কোড ব্যবহারের সুবিধা কী? [বাকাশিবো-২০০৯]
উত্তর: এটি এমনই এক ধরনের কোড, যাতে একটি সংখ্যা হতে পরবর্তী সংখ্যায় যেতে মাত্র একটি বিটের পরিবর্তন ঘটে।

CamScanner

অধ্যায়-৪
লজিক গেইটের ধারণা
(Concepts of Logic Gates)

৪.১ লজিক গেইটের সংজ্ঞা (Definition of logic gates) ঃ

যে ইলেকট্রনিক সার্কিট যুক্তিমূলক সিদ্ধান্ত তৈরি করতে পারে তাকে ϕ লজিক গেইট বলে।

লজিক সার্কিট বলতে বুঝায় এমন একটি বিশেষ ধরনের সার্কিট, যা AND, OR, NOT ইত্যাদির উপর ভিত্তি করে একটি নির্দিষ্ট লজিক্যাল ফাংশন (Logical function) সম্পাদন করে। সাধারণত দুটি ভিন্ন ধরনের ভোল্টেজ লেভেলের উপর কাজ করে যেমন– উচ্চ (High) এবং নিম্ন (Low) এবং এটিই বাইনারি লজিক সার্কিট হিসেবে বর্ণিত হয়ে থাকে। কম্পিউটারে বিভিন্ন ধরনের নির্দেশাবলি বহন এবং অঙ্কের হিসাব সম্পাদনে এই ধরনের বাইনারি সার্কিট ব্যাপকভাবে ব্যবহৃত হয়ে থাকে।

প্রধান তিনটি মৌলিক লজিক গেইট হচ্ছে 'NOT' গেইট, 'OR' গেইট, 'AND' গেইট। এই তিনটি গেইটের সমন্বয়ে বিভিন্ন ডিজিটাল গেইট তৈরি করা হয়ে থাকে। শুধু গেইট বলতে বুঝায় একটি লজিক সার্কিট, যার একটি অথবা আরও বেশি ইনপুট সিগন্যাল আছে কিন্তু আউটপুট সিগন্যাল থাকে মাত্র একটি।

যে নিয়মের সাহায্যে কয়েকটি সংখ্যা বা রাশিকে পরিচালনা করা যায়, তাই লজিক। বুলিয়ান অ্যালজেব্রার সত্যি এবং মিথ্যেকে বাইনারি 1 ও 0 দিয়ে পরিবর্তন করে কম্পিউটারের অঙ্ক কষার সমস্ত গাণিতিক সমস্যা বুলিয়ান অ্যালজেব্রার সাহায্যে সমাধান করা সম্ভব হয়।

৪.২ লজিক গেইটের প্রকারভেদ (Classification of logic gate) ঃ

১। **মৌলিক গেইট (Basic gate) বা, Fundamental gate ঃ**
   (a) AND গেইট (AND gate)
   (b) OR গেইট (OR gate)
   (c) NOT গেইট (NOT gate)

২। **ইউনিভার্সেল গেইট (Universal gate) ঃ**
   (a) NAND গেইট (NAND gate)
   (b) NOR গেইট (NOR gate)

৩। **এক্সক্লুসিভ গেইট (Exclusive gate) ঃ**
   (a) এক্সক্লুসিভ অর গেইট (Exclusive OR gate– X-OR gate)
   (b) এক্সক্লুসিভ নর গেইট (Exclussive NOR gate– X-NOR gate)

৪.৩ অ্যান্ড, অর, নট, নর, ন্যান্ড, এক্স-অর এবং এক্স-নর গেইটের যুক্তি রাশিমালা, ট্রুথ টেবিল এবং প্রতীক (Explain logical expression, truth table and symbol of AND, OR, NOT, NOR, NAND, EX-OR and EX-NOR gates) ঃ

যে নিয়মের সাহায্যে কয়েকটি সংখ্যা বা রাশিকে পরিচালনা করা যায়, তা-ই লজিক (Logic) এবং গেইট (gate) বলতে বুঝায় একটি লজিক সার্কিট, যার একটি অথবা আরও বেশি ইনপুট সিগন্যাল থাকে কিন্তু আউটপুট সিগন্যাল থাকে মাত্র একটি।

যে-সমস্ত (ইলেকট্রনিক সার্কিট) লজিক সার্কিটের সাহায্যে (বুলিয়ান সূত্র অনুযায়ী) বাইনারি অঙ্ক করা হয়, তাদের নাম Digital gate সার্কিট। লজিক সার্কিট সাধারণত দু'টি ভিন্ন ধরনের ভোল্টেজ লেভেলের উপর কাজ করে (যেমন– High এবং Low)। তাই একে বাইনারি লজিক সার্কিট হিসেবে গণ্য করা হয়ে থাকে। প্রথম প্রজন্মের কম্পিউটারে এই গেইটগুলো রিলে (Relay) যন্ত্রের সাহায্যে তৈরি করা হতো। আধুনিক আইসি প্রযুক্তিতে সবরকম ডিজিটাল গেইট, আইসি হিসেবে তৈরি করা হয়। যেমন– বুলিয়ান OR অপারেশন করার জন্য ডিজিটাল OR গেইট আইসি পাওয়া যায়। আবার বুলিয়ান AND, NOT ইত্যাদি অপারেশনের জন্য পাওয়া যায় ডিজিটাল AND, NOT গেইট আইসি।

CamScanner

যে-কোনো ডিজিটাল পদ্ধতির মূলে রয়েছে মাত্র তিনটি মৌলিক লজিক গেইট। এই তিনটি গেইট হলো–

(ক) অর গেইট (OR gate) (খ) অ্যান্ড গেইট (AND gate) (গ) নট গেইট (NOT gate)।

আবার একটি সাধারণ OR গেইটের আউটপুটে একটি NOT গেইট যুক্ত করে NOTOR বা NOR গেইট পাওয়া যায়। অনুরূপভাবে, OR গেইট থেকে Exclusive OR বা EX-OR গেইট তৈরি করা যায়। তেমনি AND গেইটের আউটপুটে একটি NOT গেইট যুক্ত করলে NOTAND অথবা সংক্ষেপে NAND গেইট পাওয়া যায়। অতএব, আমরা লজিক্যাল গেইটগুলোকে নিম্নরূপভাবে ভাগ করতে পারি; যথা–

১। AND gate
২। OR gate
৩। NOT gate
৪। NAND gate (NOT + AND)
৫। NOR gate (NOT + OR)
৬। EX-OR gate (Exclusive OR gate)
৭। EX-NOR gate (Exclusive NOR gate)।

এখানে গেইটগুলোর বর্ণনা দেয়া হলো ঃ

১। **অ্যান্ড গেইট (AND gate) ঃ** AND gate বলতে বুঝায় এমন একটি সার্কিট, যার দুই বা ততোধিক ইনপুট এবং একটিমাত্র আউটপুট থাকে এবং আউটপুট হাই (High) হবে, যদি সবগুলো ইনপুট একসাথে হাই (High) হয়। হাই ভোল্টেজের জন্য বাইনারিতে '0' এবং হাই ভোল্টেজের জন্য '1' ধরা হয়।

২। **অর গেইট (OR gate) ঃ** OR gate বলতে বুঝায় এমন একটি সার্কিট, যার দুই অথবা ততোধিক ইনপুট এবং একটিমাত্র আউটপুট থাকবে এবং যদি সমস্ত ইনপুট লো (Low) হয়, তবে আউটপুটও লো হবে।

৩। **নট গেইট (NOT gate) ঃ** NOT gate-কে ইনভার্শন গেইটও বলা হয়। বুলিয়ান ইনভার্শন বা কমপ্লিমেন্ট করার জন্য কম্পিউটারে ব্যবহার করা হয় NOT গেইট। এই গেইটটিতে কেবলমাত্র একটি ইনপুট ও একটি আউটপুট থাকে এবং আউটপুট হাই হবে, যদি ইনপুট লো হয়। আবার সার্কিটের আউটপুট লো হবে, যদি ইনপুট হাই হয়। সুতরাং একটি NOT গেইটের ইনপুটে যে ডিজিটাল তথ্যই দেয়া হউক না কেন, আউটপুটে সর্বদা কমপ্লিমেন্ট বা ইনভার্টেড তথ্য পাওয়া যায়। তাই অনেক সময় NOT গেইটকে ইনভার্টারও (Inverter) বলা হয়ে থাকে।

৪। **ন্যান্ড গেইট (NAND gate) ঃ** AND গেইটের আউটপুটে NOT লাগিয়ে NAND গেইট তৈরি করা হয়। ফলে বিভিন্ন পরিস্থিতিতে একটি সাধারণ AND গেইট যা আউটপুট দিত, তার ঠিক উল্টো (কমপ্লিমেন্ট) আউটপুট পাওয়া যায় একটি NAND গেইট থেকে।

৫। **নর গেইট (NOR gate) ঃ** একটি সাধারণ OR গেইটের আউটপুটে যদি একটি NOT gate জুড়ে দেয়া হয়, তবে যা পাওয়া যায়, সেটি হলো NOR গেইট। আউটপুটে NOT গেইট লাগানোর দরুন একটি OR গেইটের কোন বিশেষ ইনপুটের ক্ষেত্রে যা আউটপুট হয়, NOR গেইটের ক্ষেত্রে ঠিক তার উল্টো বা কমপ্লিমেন্ট আউটপুট পাওয়া যায়। NOR গেইটের ইনপুটে দুই অথবা তার চেয়ে বেশি সিগন্যাল থাকে কিন্তু কেবলমাত্র একটি আউটপুট সিগন্যাল থাকে। আউটপুট হাই (High) হবে যদি সমস্ত ইনপুট লো হয়।

৬। **Exclusive OR gate (Ex-OR gate) ঃ** Ex-OR gate-কে X-OR gate-ও বলা হয়। X-OR gate-এর বিজোড় সংখ্যক ইনপুট হাই হলেই আউটপুট হাই হয়, অন্যথায় এর আউটপুট লো হয়।

৭। **Exclusive NOR gate (Ex-NOR gate) ঃ** কোনো লজিক গেটের জোড় সংখ্যক ইনপুট হাই (High) হলে আউটপুট High হয়, অন্যথায় আউটপুট লো (Low) হয়, তাকে X-NOR gate বলে।
X-OR gate-এর বিপরীত X-NOR gate।