یونٹ 3

ڈیجیٹل لاجک اینڈ ڈیزائن

اس باب کا مطالعہ کرنے کے بعد طلبہ اس قابل ہو جائیں گے وہ:

- بولین فنکشنز اور ان کے آپریشنز جیسے کہ AND اور OR کو سمجھ سکیں۔
- متغیرات (Variables) اور Boolean Operators کا استعمال کرتے ہوئے بولین ایکسپریشنز بنا سکیں۔
- عام Boolean Identities اور Boolean Simplification Procedures سے تعلق جوڑ سکیں۔
- بولین الجبرا میں Duality کے تصور کو سمجھ سکیں۔
- ذیلی موضوعات جیسے اینالاگ (Analog) اور ڈیجیٹل (Digital) سگنلز کا تعارف کرا سکیں۔
- مختلف قسم کے لاجک گیٹس (Logic Gates) اور ان کے افعال متعارف کرا سکیں۔
- منطقی اظہار (Logical Expressions) کے آپریشنز کے لیے Truth Tables بنا سکیں۔
- K-Maps کا استعمال کرتے ہوئے بولین ایکسپریشنز کو کم سے کم (Minimize) کر سکیں۔
- ڈیجیٹل سسٹم کے لاجک ڈایاگرامز متعارف کرا سکیں۔
- Half-Adder اور Full-Adder سرکٹس کا تجزیہ اور ڈیزائن کر سکیں۔

تعارف

اس باب میں بولین فنکشن، لاجک اور ڈیجیٹل لاجک، اینالاگ اور ڈیجیٹل سگنل کے درمیان فرق پر تبادلۂ خیال کیا جائے گا۔ ہم کئی قسم کے گیٹس پر بھی تبادلہ خیال کریں گے جو ان کی ٹروتھ ٹیبل اور ڈیجیٹل ڈیوائسز کی تشکیل کرتے ہیں جن میں ہاف ایڈرز اور فل ایڈرز شامل ہیں۔

اس باب کی تکمیل پر مندرجہ ذیل ہدایات کو کرنا اور سمجھنا آسان ہونا چاہیے:

- بولین ایکشنز بنائیں اور انھیں آسان بنائیں۔
- ٹروتھ ٹیبل بنائیں۔
- ڈیجیٹل لاجک کے بنیادی تصورات سیکھیں۔

3.1 ڈیجیٹل سسٹم کے بنیادی تصورات

ڈیجیٹل سسٹم آج کے الیکٹرانکس اور کمپیوٹنگ سسٹم کی ریڑھ کی ہڈی ہے۔ یہ ڈیجیٹل معلومات کو بائنری یعنی 0 یا 1 کی شکل میں پراسیس کرتے ہیں اور حساب کتاب کے آلات جیسے کیلکولیٹر اور کمپیوٹرز میں استعمال ہوتے ہیں۔

3.1.1 اینالاگ سگنل کیا ہے

اینالاگ سگنل وہ سگنل ہیں جو وقت کے ساتھ بلا رکاوٹ اور تسلسل سے تبدیل ہوتے رہتے ہیں۔ مخصوص رینج کے درمیان ان کی کوئی بھی

ویلیو ہوسکتی ہے۔مثال کے طور پر اس میں آواز کے سگنل (بولنا)، جسم کا درجۂ حرارت اور ریڈیو ویو سگنل شامل ہیں۔

ڈیجیٹل سگنل ایسے سگنل ہیں جن کی صرف دو قیمتیں 0 اور 1 ہوتی ہیں۔ان کا استعمال ڈیجیٹل الیکٹرانکس اور کمپیوٹنگ سسٹم میں استعمال کیا جاتا ہے۔اینالاگ ٹو ڈیجیٹل کنورٹر (ADC) اور ڈیجیٹل ٹو اینالاگ کنورٹرز (DAC) آج ترقی کے دور میں اہم آپریشنز ہیں جو سگنلز کی ترسیل اور کنٹرول کو ممکن بناتے ہیں۔

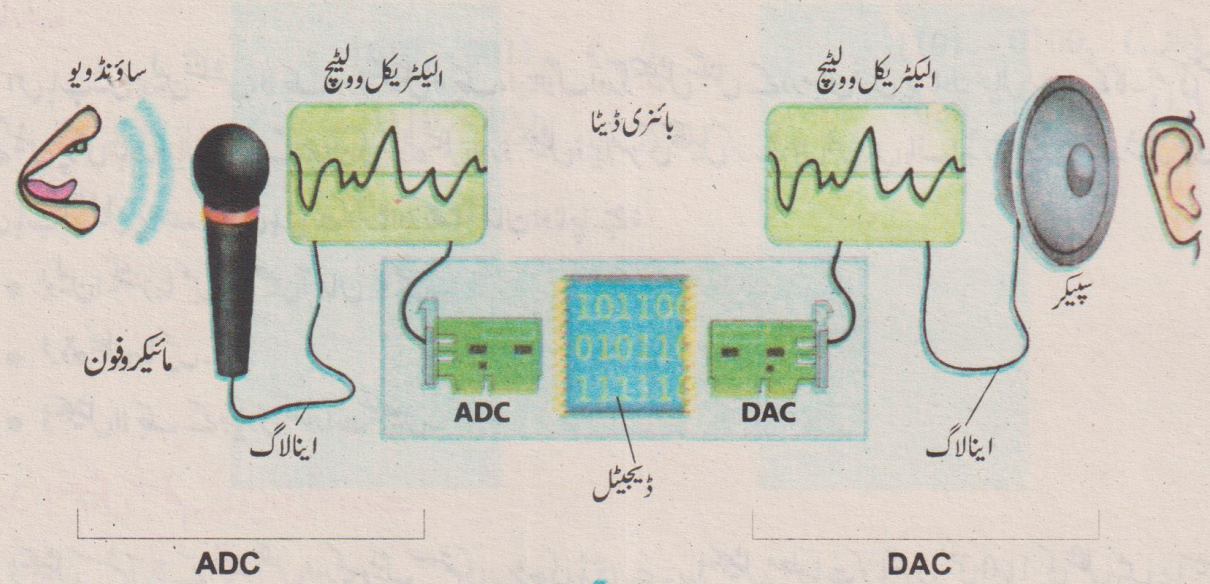
ڈیجیٹل سگنل	اینالاگ سگنل
غیر مسلسل	مسلسل
محدود قیمتیں (0 یا 1)	لامحدود ممکنہ قیمتیں
مثال: کمپیوٹرز میں بائنری ڈیٹا	مثال کے طور پر: آواز کی لہریں

اینالاگ ٹو ڈیجیٹل کنورژن (ADC):

اے ڈی سی موصول ہونے والے اینالاگ سگنلز کو ڈیجیٹل سگنلز میں تبدیل کرتا ہے جو الگ الگ ہوتے ہیں اور کمپیوٹرائزڈ ڈیوائسز جیسے کمپیوٹرز اور اسمارٹ فونز میں آسانی سے پروسیس کیے جاسکتے ہیں۔

ڈیجیٹل ٹو اینالاگ کنورژن (DAC):

ڈی اے سی وہ تبدیلی ہے جس کے ذریعے ڈیجیٹل سگنلز کو اینالاگ سگنلز میں تبدیل کیا جاتا ہے۔یہ تبدیلی انسان کے لیے معلومات کو سمجھنے کو ممکن بنانے کے لیے ضروری ہے جیسا کہ تصویر 3.1 میں دکھایا گیا ہے۔



تصویر 3.1

اے ڈی سی اور ڈی اے سی تبدیلی: اس کی ضرورت کیوں؟

ڈیجیٹل سے اینالاگ کی تبدیلی اور اینالاگ سے ڈیجیٹل میں تبدیلی اہم ہے کیونکہ یہ ڈیٹا کی پروسیسنگ، اسٹوریج اور ترسیل کو مؤثر بناتی ہے۔ڈیجیٹل سگنلز شور (Noise) اور سگنل کے خراب ہونے سے بہت کم متاثر ہوتے ہیں، اس لیے یہ طویل فاصلے تک معلومات کی منتقلی اور

ان کو ذخیرہ کرنے کے لیے زیادہ موزوں ہوتے ہیں۔

مثال کے طور پر: صوتی لہریں

آیئے! ہم ایک مخصوص صورتحال کا اندازہ لگاتے ہیں جہاں ایک شخص مائیکروفون میں کچھ بول رہا ہے جب کہ دوسرا شخص اسپیکر کے ذریعے آواز سن رہا ہے جیسا کہ اوپر شکل میں دکھایا گیا ہے۔

(1) **مائیکروفون (ADC):** جب آپ مائیکروفون میں بات کرتے ہیں تو آپ کی آواز کی لہریں (اینالاگ سگنل) پیدا کرتی ہیں جنھیں سسٹم کیچ کرتا ہے۔ یہ مائیکروفون کے ذریعہ ADC کا استعمال کرتے ہوئے مندرجہ بالا صوتی لہروں کو ڈیجیٹل شکل میں تبدیل کرتا ہے۔ آخر میں یہ ڈیجیٹل معیار کی گراوٹ کے بغیر طویل فاصلے پر منتقل کیا جاسکتا ہے۔

(2) **اسپیکر (DAC):** آخر میں ڈیجیٹل سگنلز کو پھر DAC کی مدد سے اینالاگ سگنلز میں واپس تبدیل کر دیا جاتا ہے۔ اس کے بعد اسپیکر اینالاگ آوازوں کو لہروں میں تبدیل کرتے ہیں تاکہ آپ دوسرے شخص کی آواز کو اس طرح سن سکیں جیسے آپ ان سے بات کر رہے ہوں۔

کیا آپ جانتے ہیں؟

اینالاگ سگنلز کو کبھی کبھی ڈیجیٹل سگنلز میں تبدیل کیا جاتا ہے جسے اینالاگ ٹو ڈیجیٹل تبدیلی یا ADC کہا جاتا ہے۔ یہ اینالاگ معلومات جیسے موسیقی کو ڈیجیٹل گیجٹس کے ذریعے ریکارڈ کرنے اور اس پر کام کرنے کے قابل بناتا ہے۔

3.1.2 ڈیجیٹل لاجک کے بنیادی اصول

ڈیجیٹل لاجک ڈیجیٹل سسٹم کی بنیاد ہے اس میں بائنری نمبروں کا استعمال، جو 0 یا 1 ہیں، شامل ہے جو معلومات کی نمائندگی اور اس پر کام کرنے کے لیے استعمال ہوتے ہیں۔ ڈیجیٹل لاجک کے سرکٹ مختلف آپریشنز کو انجام دینے کے لیے ان بائنری اقدار کا استعمال کرتے ہیں اور یہ کمپیوٹرز اور بہت سے دوسرے الیکٹرانک آلات کے آپریشن میں ضروری ہیں۔

ڈیجیٹل سرکٹس میں، دو حالتیں ہوتی ہیں جو 0 اور 1 کے طور پر جانی جاتی ہیں اور یہ مختلف وولٹیج کی سطحوں کی نمائندگی کرتی ہیں۔ عام طور پر، زیادہ وولٹیج، مثلاً 5 وولٹ، بائنری 1 کے طور پر اور کم وولٹیج، جیسے 0 وولٹ، بائنری 0 کے طور پر سمجھا جاتا ہے۔ ان وولٹیج کی سطحوں کو "لاجک لیول" کہا جاتا ہے، جو ڈیوائسز کو آن یا آف کرنے کے لیے ضروری ہوتی ہیں۔ لاجک لیول کا استعمال ان طریقوں کو بیان کرنے کے لیے بھی کیا جاتا ہے جن کے ذریعے ڈیجیٹل سرکٹس کام کرتے ہیں، آپریشن کرتے ہیں اور معلومات کو تبدیل کرتے ہیں۔

3.2 بولین الجبرا اور لاجک گیٹس

بولین الجبرا ریاضی کی ایک شاخ ہے جو لاجک اور علامتی الجبرا سے متعلق ہے جس میں دو قیمتیں یعنی درست اور غلط کا استعمال کیا جاتا ہے۔ یہ ڈیجیٹل سرکٹس کی ایک لازمی شاخ ہے اور یہ سرکٹوں کے تجزیے اور ڈیزائن کی بنیاد ہے۔ یہاں اس سیکشن میں بولین فنکشن اور ایکسپریشنز، لاجک گیٹس کے کام اور فنکشن، ٹروتھ ٹیبل اور لاجک کی ڈایاگرامز کی تعمیر اور تشخیص کا موضوع ہے۔

3.2.1 بولین فنکشن اور ایکسپریشنز

بائنری اقدار کا استعمال بولین فنکشن اور بولین ایکسپریشنز میں ویری ایبلز کے مابین تعلق کی وضاحت کرنے کے لیے کیا جاتا ہے۔ یہ

''اینڈ (AND)'' یا ''آر (OR)'' فنکشن کا استعمال کرتے ہوئے بنائے گئے ہیں اور ڈیجیٹل سرکٹوں کو بہتر بنانے کے لیے سے استعمال کیے جاتے ہیں۔

3.2.1.1 بائنری ویری ایبلز اور لاجک آپریشنز

بائنری ویری ایبل میں صرف دو قیمتیں 0 اور 1 ہو سکتی ہیں، لاجک آپریشنز ان بائنری ویرایبلز کی پروسیسنگ کے لیے بولین الجبرا میں نافذ کردہ ابتدائی آپریشنز ہیں۔ بنیادی لاجک آپریشنز ''اینڈ (AND)''، ''آر (OR)'' اور ''ناٹ (NOT)'' ہیں۔

AND آپریشن:

''اینڈ (AND)'' بنیادی لاجک آپریٹر ہے جو بولین الجبرا میں استعمال ہوتا ہے۔ اس قسم کے فنکشن کو دو بائنری ان پٹ کی ضرورت ہوتی ہے جو بائنری شکل میں ایک ہی آؤٹ پٹ دیتا ہے۔ علامت '.' آپریشن (AND) کے لیے استعمال کی جاتی ہے اس کا مطلب ہے کہ AND آپریشن کی آؤٹ پٹ صرف اس وقت 1 ہوگی جب دونوں ان پٹ 1 ہوں گی۔ اگر ان پٹ میں سے کسی میں 0 قیمت ہوئی تو اس کا رزلٹ 0 ہو جائے گا۔

مثال: دو بائنری ویری ایبلز پر غور کریں:

A= 1 (True)

B= 0 (False)

ان ویرایبلز کے AND آپریشن کو ریاضیاتی طور پر یوں لکھا جا سکتا ہے:

P= A . B

اب

A = 1 , B = 0

اس لیے فنکشن P کا رزلٹ 0 یعنی False ہوگا۔

ٹروتھ ٹیبل:

ان پٹ ویری ایبلز کے کی تمام ممکن ان پٹس پر AND آپریشن کی فعالیت کو ظاہر کرنے میں ایک ٹروتھ ٹیبل بہت مفید ہے۔ AND آپریشن کے لیے ٹروتھ ٹیبل درج ذیل ہے:

A	B	A AND B (P)
0	0	0
0	1	0
1	0	0
1	1	1

ٹیبل 3.1: AND آپریشن کی وضاحت کے لیے ٹروتھ ٹیبل

- اگر A اور B دونوں 0 یعنی Off ہیں، جو صفر کے برابر ہے تو مطلوبہ آؤٹ پٹ (P) بھی Off ہوگا۔
- اسی طرح اگر A '0' اور B '1' ہے تو مطلوبہ آؤٹ پٹ (P) Off ہوگا۔
- اسی طرح اگر B '0' اور A '1' ہے تو مطلوبہ آؤٹ پٹ (P) Off ہوگا۔
- لیکن جب B '1' اور A بھی '1' ہے تو مطلوبہ آؤٹ پٹ (P) ON ہوگا۔

OR آپریشن:

OR آپریشن بولین الجبرا میں دوسرا بنیادی لاجیکل آپریٹر ہے۔ یہ بھی دو بائنری ویری ایبلز کا ایک فنکشن ہے اور ایک واحد بائنری ویری ایبل فنکشن کی آؤٹ پٹ ہے۔ ٹیبل 3.2 کے مطابق، OR آپریشن (1) آؤٹ پٹ پیدا کرتا ہے جب کم از کم ایک ان پٹ 'درست' ہوتا ہے۔ آؤٹ پٹ صرف اس وقت 0 ہوتا ہے جب دونوں ان پٹ 0 ہوتے ہیں۔ اس وجہ سے آؤٹ پٹ 'غلط' (0) صرف اس صورت میں ہوتا ہے جب دونوں ان پٹ 'غلط' ہوں۔

مثال: دو بائنری ویری ایبلز پر غور کریں:

A= 1 (True)

B= 0 (False)

ان ویری ایبلز کے لیے OR آپریشن کو ریاضیاتی طور پر یوں لکھا جا سکتا ہے:

$$P = A + B$$

اب

$$A = 1 \quad , \quad B = 0$$

اس لیے فنکشن P کا رزلٹ 1 یعنی True ہوگا۔

اس کا مطلب یہ ہے کہ اگر آپ کے پاس A= 1 (True) اور B= 0 (False) ہے تو، OR گیٹ کا نتیجہ درج ذیل 1=(P) ہوگا جو OR گیٹ کا بولین ایکسپریشن ہے:

ایک ٹروتھ ٹیبل اس بات کی بہتر تفہیم کے لیے مفید ہے کہ OR آپریشن کو کس طرح منظم کیا جاتا ہے اور ان پٹ ویری ایبلز کی تمام ویلیوز کے لیے OR کے اطلاق کا نتیجہ کیا ہے۔ ذیل میں OR آپریشن کے لیے ٹروتھ ٹیبل ہے۔

A	B	A OR B (P)
0	0	0
0	1	1
1	0	1
1	1	1

ٹیبل 3.2: OR آپریشن کی وضاحت کے لیے ٹروتھ ٹیبل

وضاحت:

- اگر A اور B دونوں 0 یعنی Off ہیں، جو صفر کے برابر ہے تو مطلوبہ آؤٹ پٹ (P) بھی Off ہوگا۔
- اگر A '0' اور B '1' ہے تو مطلوبہ آؤٹ پٹ (P) 1 ہوگا۔
- اگر B '0' اور A '1' ہے تو مطلوبہ آؤٹ پٹ (P) 1 ہوگا۔
- اسی طرح جب B '1' اور A بھی '1' ہے تو مطلوبہ آؤٹ پٹ (P) 1 ہوگا۔

کیا آپ جانتے ہیں؟

بائنری لاجک میں 1+1، 2 کے برابر نہیں ہوتا بلکہ یہ لاجیکل آپریشنز میں 1 کے برابر ہی ہوتا ہے کیوں کہ اگر اِن پٹ میں کوئی ایک قیمت بھی 1 ہے تو OR آپریشن ہمیشہ 1 ہی ریٹرن کرے گا۔

NOT آپریشن:

NOT آپریشن بنیادی بولین الجبرا آپریشنز میں سے ایک ہے جو ایک بائنری ویری ایبل لیتا ہے اور صرف قیمت کی نفی کرتا ہے۔ اگر ان پٹ 1 ہے تو آؤٹ پٹ 0 ہوگی اور اگر ان پٹ 0 ہے تو آؤٹ پٹ 1 ہوگی۔

مثال: ایک بائنری ویری ایبل پر غور کریں:

A= 1 (True)

اس ویری ایبل کے لیے NOT آپریشن ریاضیاتی طور پر لکھا جا سکتا ہے:

P = -A یا P = A

اس مثال میں:

P = 0

اس کا مطلب یہ ہے کہ اگر آپ کے پاس A= 1 (True) ہے تو NOT آپریشن کا نتیجہ 0 (False) ہو جائے گا۔

ٹروتھ ٹیبل:

ذیل میں NOT آپریشن کے لیے ٹروتھ ٹیبل ہے:

A	NOT A
0	1
1	0

ٹیبل 3.3: NOT آپریشن کی وضاحت کے لیے ٹروتھ ٹیبل

وضاحت:

جب اِن پٹ (A) ، 0 ہوتا ہے تو آؤٹ پٹ (P) ، 1 ہوتا ہے کیوں کہ NOT آپریشن اِن پٹ کیے گئے ویری ایبل کے منفی یا

مخالف قیمت دیتا ہے۔

یہ آپریشن ڈیجیٹل لاجک ڈیزائن میں زیادہ پیچیدہ لاجک فنکشنز بنانے اور ڈیجیٹل سرکٹس کی فعالیت کی تصدیق کرنے کے لیے اہم ہے۔

3.2.1.2 بولین فنکشن کی تعمیر

بولین فنکشن الجبری بیانات ہیں جو بائنری ویری ایبلز اور لاجک کی کارروائیوں کے درمیان تعلقات کی وضاحت کرتے ہیں۔ یہ فنکشن ڈیجیٹل لاجک ڈیزائن کے لیے خاص طور پر اہم ہیں اور مختلف ڈیجیٹل سرکٹس کی تشکیل میں استعمال ہوتے ہیں جو موجودہ کمپیوٹرز، موبائل فونز اور یہاں تک کہ سادہ کیلکولیٹر کی بنیاد ہیں۔

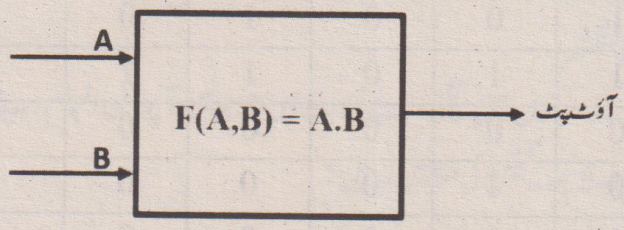
بولین فنکشنز کو سمجھنا

بولین فنکشن ایک ایسا فنکشن ہے جس میں ایک یا ایک سے زیادہ بائنری اِن پُٹس ہوتی ہیں اور صرف ایک ہی بائنری آؤٹ پٹ ہوتی ہے۔ ان پٹ اور آؤٹ پٹ میں صرف دو قدریں ہوسکتی ہیں، فالس کو 0 اور ٹرو کو 1 کی قیمت تفویض کی گئی ہے۔ بولین فنکشن کی تعمیر سے بنیادی لاجکس (NOT، OR،AND) کی کارروائیوں کو استعمال کرتے ہوئے کی جاتی ہے تاکہ صحیح آؤٹ پٹ حاصل کرنے کے لیے ان پٹ کو صحیح طریقے سے مربوط کیا جاسکے۔

مثال 1: سادہ بولین فنکشن

دو اِن پٹ، A اور B کے ساتھ بولین فنکشن پر غور کریں۔ ہم ایک فنکشن F تشکیل دے سکتے ہیں جو AND آپریشن کی نمائندگی کرتا ہے:

$$F(A,B) = A \cdot B$$



شکل 3.2: سادہ بولین فنکشن

اوپر دکھایا گیا ڈایاگرام ایک بنیادی ڈیجیٹل سرکٹ کو ظاہر کرتا ہے، یہ سرکٹ ایک اینڈ گیٹ ہے۔ یہ باکس F(A,B)=A.B فنکشن کی علامت ہے اس باکس میں دو اِن پٹ A اور B ہیں۔ A اور B اگر دونوں 1 ہیں تو یہ 1 کے برابر ہوگا۔ کسی بھی دوسرے معاملے میں، آؤٹ پٹ صرف 0 ہوگا۔ ان پٹ کو باکس کے داخلی دروازے پر دکھایا گیا ہے جبکہ آؤٹ پٹ کو بلاک کے باہر نکلنے پر دکھایا گیا ہے۔ اس فنکشن کے لیے ٹروتھ ٹیبل مندرجہ ذیل ہے:

A	B	F (A , B)
0	0	0
0	1	0

1	0	0
1	1	1

ٹیبل 3.4: F(A,B)=A.B کے لیے ٹروتھ ٹیبل

مثال 2: اب، آیئے ہم تین ان پٹ A، B اور C کے ساتھ ایک زیادہ پیچیدہ بولین فنکشن تشکیل دیتے ہیں:

$$F(A,B,C)=A.B+\overline{A}.C$$

یہ فنکشن ایک ہی وقت میں AND، OR اور NOT کا استعمال کرتا ہے۔ اس فنکشن کے لیے ٹروتھ ٹیبل مندرجہ ذیل ہے:

وضاحت:

- پیرامیٹرز A، B اور C کے مندرجہ ذیل مثال میں ان پٹ کالموں کے طور پر شامل ہیں۔
- دو ویری ایبلز A اور B کے درمیان AND آپریشن کے نتائج کالم A.B میں پیش کیے گئے ہیں۔
- کالم $\overline{A}$، NOT آپریشن کے لیے پیش کیا گیا ہے۔
- کالم $\overline{A}.C$ پانچویں کالم اور تیسرے کالم میں اقدار کے درمیان AND آپریشن کا نتیجہ ظاہر کرتا ہے۔
- آخری کالم F (A, B, C) بولین فنکشن $(A . B) + (\overline{A} . C)$ کی آؤٹ پٹ دکھاتا ہے۔

A	B	C	A.B	$\overline{A}$	$\overline{A}$.C	F (A , B, C)
0	0	0	0	1	0	0
0	0	1	0	1	1	1
0	1	0	0	1	0	0
0	1	1	0	1	1	1
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	1	0	0	1
1	1	1	1	0	0	1

ٹیبل 3.4: F(A,B)=A.B کے لیے ٹروتھ ٹیبل

کمپیوٹر میں استعمال:

مختلف آپریشنز کے لیے کمپیوٹرز میں بولین فنکشن کے بہت سے استعمال ہیں۔ یہاں ان کے استعمال کی کچھ مثالیں ہیں:

- ریاضی کی کارروائیاں: سی پی یو کے ارتھمیٹک اینڈ لاجیکل یونٹ (ALU) میں بولین فنکشن، جمع، تفریق، ضرب اور یہاں تک کہ تقسیم کرنے کا انتظام کرتے ہیں۔
- ڈیٹا پروسیسنگ: بولین فنکشن میموری اور اسٹوریج ڈیوائسز میں بائنری ڈیٹا پر کارروائی کرنے کے لیے استعمال ہوتے ہیں، موثر ڈیٹا پروسیسنگ اور بازیافت کو یقینی بناتے ہیں۔

- **کنٹرول لاجک:** بولین فنکشن کمپیوٹرز میں لاگو ہوتے ہیں جو مربوط انداز میں کام کرنے کے لیے سسٹم کے آپریشن کے مختلف حصوں کو کنٹرول کرتے ہیں۔

کیا آپ جانتے ہیں؟

بولین فنکشن ہمارے روزمرہ کے آلات جیسے سیل فون اور کیلکولیٹر میں بھی موجود ہیں:

سیل فون: سیل فون کی پروسیسنگ میں، جب آپ کوئی نمبر ڈائل کرتے ہیں، یا فون پر ایک بٹن دباتے ہیں تو، بولین فنکشن ان ان پٹ کو صحیح یا غلط کے طور پر ایڈریس کرتا ہے اور آؤٹ پٹ فراہم کرتا ہے۔

کیلکولیٹر: بنیادی کیلکولیٹر میں بولین فنکشن استعمال ہوتے ہیں، جب آپ اسے نمبروں اور ان پر انجام دیے جانے والے آپریشنز کے ساتھ فیڈ کرتے ہیں تو، بولین لاجک صحیح نتائج تک پہنچنے کے لیے استعمال ہوتی ہے۔

سرگرمی

غور کریں کہ آپ روزانہ کی بنیاد پر اپنے سیل فون یا کیلکولیٹر کے ساتھ کیا کرتے ہیں۔ کیا آپ ان سرگرمیوں میں فرق کر سکتے ہیں جن کے لیے لاجک کے انتخاب کی ضرورت ہوتی ہے، جیسے اپنے اسمارٹ فون کو ان لاک کرنے کے لیے پاسورڈ درج کرنا یا ریاضی کا مسئلہ حل کرنا؟ اپنے گروپ کے ممبروں سے پوچھیں کہ پس منظر میں بولین فنکشن کو کس طرح استعمال کیا جا سکتا ہے۔

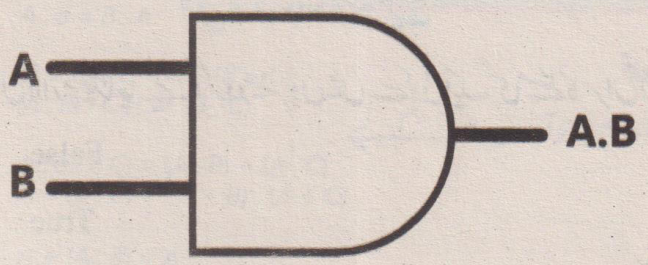
کیا آپ جانتے ہیں؟

جارج بول، ایک ریاضی دان، جس نے بولین الجبرا ایجاد کیا، 1815ء میں لنکن، انگلینڈ میں پیدا ہوا۔ ان کے کام نے مستقبل کے ڈیجیٹل انقلاب اور کمپیوٹر سائنس، مستقبل کی مختلف ٹیکنالوجیز کی بنیاد رکھی۔

3.2.2 لاجک گیٹس اور ان کے فنکشنز

لاجک گیٹس الیکٹرانک ڈیوائسز کے سرکٹوں میں لگنے والے آلات ہیں جو بولین آپریشن انجام دیتے ہیں۔ ہر قسم کا لاجک گیٹ بنیادی بولین آپریشن سے مطابقت رکھتا ہے۔ لاجک گیٹس کی مثالیں یہ ہیں:

AND گیٹ **AND** فنکشن کو نافذ کرتا ہے۔ یہ صرف اس وقت درست ہوتا ہے جب دونوں ان پٹ درست/ٹرو (1) ہوں۔



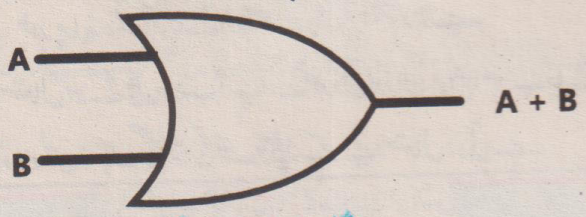
تصویر 3.3: AND گیٹ

AND گیٹ کے ساتھ ایک سادہ الیکٹرانک سرکٹ کا تصور کریں۔ اگر آپ دو سوئچ دبائیں (دونوں کو آن ہونا ضروری ہے) تو ایک لائٹ بلب آن ہو جائے گا۔

- سوئچ1: ON (True)
- سوئچ2: ON (True)
- لائٹ بلب: ON (True) کیوں کہ دونوں سوئچ ON ہیں۔

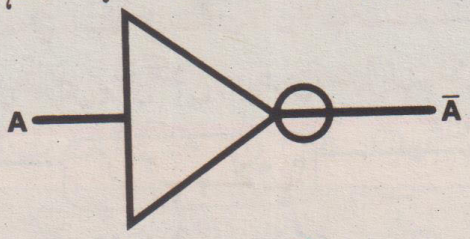
اگر دونوں میں سے کوئی ایک سوئچ بند ہے تو، لائٹ بلب بند ہو جائے گا۔

OR گیٹ: OR فنکشن کو نافذ کرتا ہے۔ جب کم از کم ایک اِن پٹ بھی درست (True) ہو تو آؤٹ پٹ True دیتا ہے۔



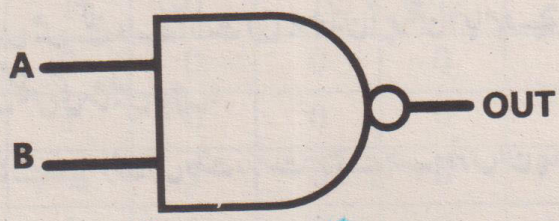
شکل 3.4: OR گیٹ

NOT گیٹ: NOT فنکشن کو نافذ کرتا ہے۔ یہ ان پٹ کے برعکس آؤٹ پٹ کرتا ہے۔ تصویر 3.5 دیکھیں۔



شکل 3.5: NOT گیٹ

NAND گیٹ: یہ گیٹ AND گیٹ کو NOT گیٹ کے ساتھ ملا کر بنایا جاتا ہے۔ یہ اس وقت True آؤٹ پٹ دیتا ہے جب کم از کم ایک اِن پُٹ False ہو۔ دوسرے لفظوں میں، یہ AND گیٹ کا الٹ ہے۔ جیسا کہ شکل 3.6 میں دکھایا گیا ہے:



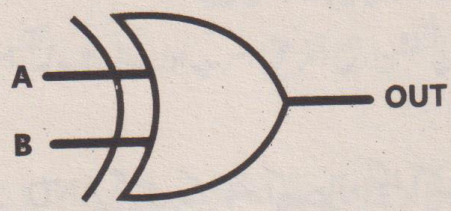
شکل 3.6: NAND گیٹ

مثال:

ایک حفاظتی نظام کا تصور کریں جہاں الارم چلنا چاہیے۔ اگر دو سینسروں میں سے کوئی ایک کسی مسئلے کا سراغ لگائے۔

- سینسر1: کوئی مسئلہ نہیں False
- سینسر2: مسئلے کا پتا چلا True
- الارم: آن (True) کیوں کہ ایک سینسر کسی مسئلے کا سراغ لگاتا ہے۔

XOR گیٹ:



شکل 3.7: XOR گیٹ

XOR گیٹ صرف اس وقت درست ہوتا ہے جب ان پٹ میں سے ایک درست ہو۔ یہ

OR گیٹ سے اس لحاظ سے مختلف ہیں کہ جب دونوں کا اِن پٹ درست (True) ہوتے ہیں تو اس کا آؤٹ پٹ بھی درست (True) ہوتا ہے۔

مثال:

ایک ایسے منظرنامے کا تصور کریں جہاں آپ یا تو ویڈیو گیمز کھیل سکتے ہیں یا ہوم ورک کر سکتے ہیں، لیکن دونوں ایک ہی وقت میں نہیں۔

- ویڈیو گیمز کھیلیں: جی ہاں (True)
- ہوم ورک: نہیں (False)
- اجازت؟ جی ہاں (True) کیوں کہ صرف ایک سرگرمی کی جا رہی ہے۔

سرگرمی

(1) آیئے! ان لاجک کے فنکشن کو سیکھنے کے لیے ایک سرگرمی کریں!

(i) **AND ایڈونچر:** 2 کے گروپس تشکیل دیں اور ہر جوڑے کو انعام جیتنے کے لیے دو شرائط پر پورا اترنے کی ضرورت ہے (جیسے دونوں نے ایک مخصوص رنگ کی قمیض پہن رکھی ہے۔)

(ii) **OR آپشنز:** تفریحی سرگرمیوں کی ایک فہرست بنائیں۔ اگر کم از کم ایک سرگرمی ممکن ہے تو، کلاس کو اضافی پلے ٹائم ملتا ہے۔

(iii) **NOT نیگیٹوز:** True/False سوالات پوچھیں اور طالب علموں کو اس کے برعکس جواب دیں۔ مثال کے طور پر "کیا آسمان سبز ہے؟"۔ طالب علموں کو اس پر "نہیں" کے نعرے لگانے چاہییں (Not True)۔

(iv) AND گیٹ کی نمائندگی کے لیے بریڈ بورڈ، بیٹری اور ایل ای ڈی لائٹس کا استعمال کرتے ہوئے ایک بنیادی سرکٹ بنائیں۔ دو سوئچوں کو آپس میں جوڑیں، اس تجربے میں یہ ان پٹ اے اور بی ہونے جا رہا ہے، جب دونوں سوئچ دبائے جائیں گے تو ہی ایل ای ڈی روشن ہو جائے گی۔

بولین الجبرا کے بنیادی اصول:

یہاں کچھ بنیادی بولین الجبرا قواعد ہیں جو آسان بنانے کے لیے استعمال ہوتے ہیں:

1. Identity Laws	$A + 0 = A$ $A \cdot 1 = A$
2. Null Laws	$A + 1 = 1$ $A \cdot 0 = 0$
3. Idempotent Laws	$A + A = A$ $A \cdot A = A$
4. Complement Laws	$A + \overline{A} = 1$ $A \cdot \overline{A} = 0$
5. Commutative Laws	$A + B = B + A$ $A \cdot B = B \cdot A$
6. Associative Laws	$(A + B) + C = A + (B + C)$ $(A \cdot B) \cdot C = A \cdot (B \cdot C)$
7. Distributive Laws	$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$ $A + (B \cdot C) = (A + B) \cdot (A + C)$
8. Absorption Laws	$A + (A \cdot B) = A$ $A \cdot (A + B) = A$
9. De Morgan's Theorems	$\overline{A + B} = \overline{A} \cdot \overline{B}$ $\overline{A \cdot B} = \overline{A} + \overline{B}$
10. Double Negation Law	$\overline{\overline{A}} = A$

3.3 بولین فنکشن کو آسان بنانا

ایک مؤثر ڈیجیٹل سرکٹ ڈیزائن کرنے میں بولین فنکشن کو آسان بنانا ایک خاص طور پر اہم عمل ہے۔ اس طرح کے فنکشن

میں گیٹس کے کم استعمال کی ضرورت ہوتی ہے جو انھیں سائز میں چھوٹا، توانائی کی بچت اور پیچیدہ کے مقابلے میں تیز تر بناتے ہیں۔ آسان بنانے کا مطلب ہے کہ فنکشن کو کم پیچیدہ بنانے کے لیے کچھ بولین الجبرا قوانین کا اطلاق کرنا۔

Example 1

ایکسپریشن کو سادہ کریں $A + \overline{A} \cdot B.$

Solution:

$A + \overline{A} \cdot B = (A + \overline{A}) \cdot (A + B)$ (Distributive Law)

$= 1 \cdot (A + B)$ (Complement Law)

$= A + B$ (Identity Law)

Example 2

ایکسپریشن کو سادہ کریں $\overline{A \cdot B} + \overline{A} \cdot B.$

Solution:

$\overline{A \cdot B} + \overline{A} \cdot B = \overline{A} + \overline{B} + \overline{A} \cdot B$ (De Morgan's Theorem)

$= (\overline{A} + \overline{B})$ Since $\overline{A}$ is already present in $(\overline{A} \cdot \overline{B})$, we can use absorption law i.e $\overline{A} + (\overline{A} \cdot B) = \overline{A}$

$= \overline{A} + \overline{B}$

Example 3

ایکسپریشن کو سادہ کریں $(A + B) \cdot (A + \overline{B})$

Solution:

$(A + B) \cdot (A + \overline{B}) = A \cdot (A + \overline{B}) + B \cdot (A + \overline{B})$ (Distributive Law)

$= A + A \cdot \overline{B} + B \cdot \overline{B}$ (Absorption Law)

$= A + A \cdot B$ (Identity Law)

$= A \cdot (1 + B)$ (Distributive Law)

$= A \cdot 1$ (Null Law)

$= A$ (Identity Law)

Example 4

ایکسپریشن کو سادہ کریں $\overline{\overline{A} + B} \cdot (A + \overline{B})$

Solution:

$\overline{\overline{A} + B} \cdot (A + \overline{B}) = (A \cdot \overline{B}) \cdot (A + \overline{B})$ (De Morgan's Theorem)

$= A \cdot \overline{B} \cdot A + A \cdot \overline{B} \cdot \overline{B}$ (Distributive Law)

$= A \cdot \overline{B} + A \cdot \overline{B}$ (Idempotent Law)

$= A \cdot \overline{B}$ (Identity Law)

3.4 لاجک ڈایاگرام بنانا

- لاجک ڈایاگرام علامتوں کے ذریعے ڈیجیٹل سرکٹ کے کام کی عکاسی کرتے ہیں جو اس کے انفرادی لاجک گیٹس سے متعلق ہیں۔
- ایک لاجک ڈایاگرام بنانے کے لیے، بولین فنکشن کے لیے ضروری لاجک گیٹس تلاش کریں۔
- کارروائیوں کو انجام دینے کے لیے گیٹس کو ترتیب دیں جیسا کہ اس کے سرکٹ کے فنکشن کے ذریعے بیان کیا گیا ہے
- ان پٹ اور گیٹس کی آؤٹ پٹ کو صحیح طریقے سے لنک کریں۔

خلاصہ کرنے کے لیے جب ڈیجیٹل سرکٹس کی تخلیق اور مطالعہ کی بات آتی ہے تو بولین الجبرا اور لاجک گیٹس کا علم اہم ہے۔ اگر طلبہ ان تصورات کو سمجھتے ہیں تو وہ اچھے ڈیجیٹل سسٹم تشکیل دے سکتے ہیں جو موثر ہوں۔

3.5 ڈیجیٹل لاجک کا اطلاق

ڈیجیٹل لاجک بہت سے الیکٹرانک سسٹم کے کام کرنے کا ایک لازمی جز ہے جو موجودہ دور میں زیرِ استعمال ہیں جیسے: کمپیوٹر، اسمارٹ فون ،اور بہت سے دوسرے ڈیجیٹل گیجٹ۔ ڈیجیٹل لاجک مختلف فنکشن کو انجام دینے کے لیے سرکٹوں کو بنانے اور بڑھانے کے لیے بہت سے طریقوں سے کام کرتی ہے۔ ڈیجیٹل لاجک کی ان دو ایپلی کیشنز میں ایڈر سرکٹس کا ڈیزائن اور فنکشن کو آسان بنانے کے لیے کارناف میپ شامل ہیں۔

3.5.1 ہاف ایڈر اور فل ایڈر سرکٹ

ایڈر سرکٹ بڑے پیمانے پر ڈیجیٹل سرکٹس میں ریاضی کے حسابات انجام دینے کے لیے استعمال ہوتے ہیں۔ ایڈر سرکٹس کی دو عام قسمیں ہیں جنھیں ہاف ایڈرز اور فل ایڈرز کے نام سے جانا جاتا ہے۔

3.5.1.1 ہاف ایڈر سرکٹ

ہاف ایڈر ایک بنیادی سرکٹری یونٹ ہے جو دو سنگل بٹ بائنری ہندسوں کو جمع کرتا ہے۔ اس میں دو ان پٹ ہیں، عام طور پر A اور B کے طور پر ظاہر ہوتے ہیں اور دو آؤٹ پٹ: مجموعہ (S) اور کیری (C)۔

ہاف ایڈر کے لیے ٹروتھ ٹیبل:

A	B	Sum (S)	Carry (C)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

ٹیبل 3.6: ہاف ایڈر کے لیے ٹروتھ ٹیبل

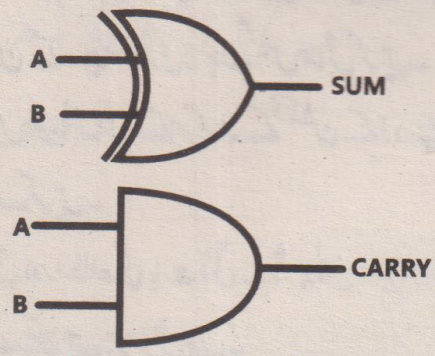
$$S = A + B$$

$$C = A \cdot B$$

یہاں پر علامت (⊕)، XOR آپریشن کی نمائندگی کرتی ہے۔ رزلٹ اس وقت ہائی ہوتی ہے جب ان پٹ میں سے صرف ایک ویلیو ہائی ہوتی ہے۔ جبکہ جب دونوں ان پٹ ہائی (1) ہوتے ہیں تو کیری آؤٹ پٹ ہائی (1) ہوتا ہے۔

بولین ایکسپریشن:

$$m = A \oplus B$$



تصویر 3.8: ہاف ایڈر سرکٹ

3.5.1.2 فُل ایڈر سرکٹ

ایک فُل ایڈر ایک زیادہ پیچیدہ سرکٹ ہے جو تین سنگل بٹ بائنری نمبر جمع کرتا ہے۔ دو مثبت بٹس اور ایک کیری بٹ جو پچھلے مثبت سے تعلق رکھتا ہے۔ اس میں تین ان پٹ ہیں، جن کو A، B اور Cin (کیری ان پٹ) کے طور پر ظاہر کیا گیا ہے، اور دو آؤٹ پٹ ہیں جنھیں مجموعہ (S) اور کیری (C) کہا جاتا ہے جس میں دونوں عددی قدریں ہیں۔ اگر ہائی ان پٹ کی تعداد طاق ہے تو مجموعی آؤٹ پٹ ہائی ہے، جبکہ اگر ہائی ان پٹ کی تعداد کم از کم 2 ہے تو کیری آؤٹ پٹ ہائی ہے۔

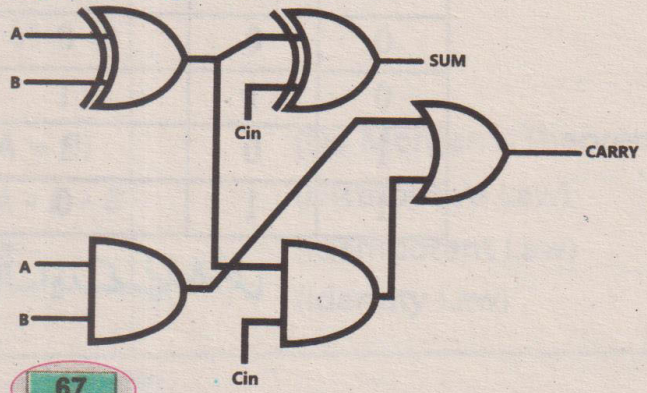
A	B	Cin	Sum(S)	Carry(Cout)
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

ٹیبل 3.9: فُل ایڈر سرکٹ

Boolean Expressions:

Sum = A⊕B⊕Cin

Carry= (A.B) + (Cin . (A⊕B))



ہاف ایڈر سرکٹ

3.5.2 کرناف میپ (K۔میپ)

کرناف میپ (کے۔میپ) ایک گرافیکل نمائندگی ہے جسے بولین الجبرا کو حل کرنے اور لاجک فنکشن کو کم سے کم کرنے کے لیے استعمال کیا جاسکتا ہے جہاں الجبری حسابات کا استعمال نہیں کیا جاتا ہے۔ یہ ایک ایسی تکنیک ہے جس میں بولین فنکشن کی درست قیمت کو نمونوں کی شناخت کو قابل بنانے اور آسان بنانے کے لیے اصطلاحی امتزاج انجام دینے کے لیے بنایا جاتا ہے۔

Minterm	Combination of Variables	Minterm Expression
0	A=0, B=0, C=0	$\overline{A}\overline{B}\overline{C}$
m1	A=0, B=0, C=1	$\overline{A}\overline{B}C$
m2	A=0, B=1, C=0	$\overline{A}B\overline{C}$
m3	A=0, B=1, C=1	$\overline{A}BC$
m4	A=1, B=0, C=0	$A\overline{B}\overline{C}$
m5	A=1, B=0, C=1	$A\overline{B}C$
m6	A=1, B=1, C=0	$AB\overline{C}$
m7	A=1, B=1, C=1	ABC

ٹیبل 3.8: ABC کے لیے ممکنہ منٹرمز

3.5.2.1 کرناف میپ کی ساخت

ایک کرناف میپ دراصل ایک میٹرکس ہے، جہاں ہر مربع ایک سیل ہے، جو ایک پوزیشن شدہ امتزاج یا ایک ان پٹ ویری ایبل سے مطابقت رکھتا ہے اور سیلز عام طور پر بولین فنکشن کے ٹروتھ ٹیبل کے حوالہ سے '1' یا '0' سے بھرے ہوتے ہیں۔ کرناف میپ کا سائز ویری ایبلز کی تعداد پر منحصر ہے:

- 2 متغیرات: 2x2 گرڈ
- 3 متغیرات: 2x4 گرڈ
- 4 متغیرات: 4x4 گرڈ کم سے کم
- 5 متغیرات: 4x8 گرڈ (بہت کم استعمال ہونے والا)

کے۔میپ میں ہر سیل ایک منٹرم کی نمائندگی کرتا ہے اور کے۔میپ کی ہر قطار میں سیلز کوڈ الفاظ میں کسی خاص پوزیشن پر صرف ایک بٹ سے مختلف ہوتے ہیں جو گرے کوڈ ترتیب کی پیروی کرتے ہیں۔

3.5.2.2 بولین الجبرا میں منٹرمز

بولین الجبرا میں، منٹرم ایک خاص پروڈکٹ کی اصطلاح ہے جس میں فنکشن کا ہر ویری ایبل یا تو 1 ہے یا پھر اس کا کمپلیمنٹ ہوتا ہے۔ اس

کی تکمیل میں ایکسپریشن کے اندر شامل ہوتا ہے۔ ہر منٹرم ویری ایبل قیمتوں کے ایک اور صرف ایک سیٹ سے مطابقت رکھتا ہے جو بولین فنکشن کو درست یا 1 کے برابر بناتا ہے۔

ویری ایبل A، B اور C کے ساتھ بولین فنکشن منٹرم نوٹیشن:

- وہ منٹرم جہاں $A = 1$، $B = 0$ اور $C = 1$ کو $A\overline{B}C$ لکھا جاتا ہے۔
- بولین فنکشن F(A,B,C) پر غور کریں:
- اس فنکشن کے لیے ممکنہ منٹرمز یہ ہیں: A، B اور C کے لیے ممکنہ منٹرمز

3.5.2.3 کارناف نقشے بنانا

K-map بنانے کے لیے، ان مراحل پر عمل کریں:

(1) سسٹم میں موجود ویرایبلز کی تعداد کی بنیاد پر ایک گرڈ بنائیں۔

(2) آیئے ہم آؤٹ پٹ اقدار کی مدد سے گرڈ کو مکمل کریں جو ٹروتھ ٹیبل میں شامل ہیں۔

(3) گرڈ میں 1 کو سائز 1، 2، 4، 8 اور اسی طرح کے سب سے بڑے ممکنہ مربعوں میں ترتیب دیں۔ ہر گروپ کے پاس ایک یا ایک سے زیادہ 1 ہونا ضروری ہے اور یہ دو کی طاقت اور انھیں مستقل قطار میں ہونا چاہیے۔

مثال: کے۔میپ کے ساتھ بولین ایکسپریشن کو آسان بنانا

1۔ ایکسپریشن: $A.\overline{B}+\overline{A}.B+A.B$

مرحلہ 1: دو ویری ایبلز A اور B کے لیے کے۔میپ گرڈ کھینچیں:

	$\overline{B} = 0$	$B = 1$
$\overline{A} = 0$	0	1
$A = 1$	1	1

مرحلہ 2: K-map کو پر کریں

اظہار کی بنیاد پر A اور B کے ہر امتزاج کے لیے آؤٹ پٹ کا تعین کریں۔

- For $A = 0$ and $B = 0$: $\overline{F} = \overline{A}.B + A.B + A.B = 1.0 + 0.0 + 0.0 = 0$
- For $A = 0$ and $B = 1$: $F = A.\overline{B} + \overline{A}.B + A.B = 0.0 + 1.1 + 0.1 = 1$
- For $A = 1$ and $B = 0$: $F = A.\overline{B} + \overline{A}.B + A.B = 1.1 + 0.0 + 1.0 = 1$
- For $A = 1$ and $B = 1$: $F = A.\overline{B} + \overline{A}.B + A.B = 1.0 + 0.1 + 1.1 = 1$

مرحلہ 3: K-map میں 1s کو گروپ کریں۔

ایکسپریشن کو آسان بنانے کے لیے 1s گروپ کے ساتھ ملائیں۔

	$\overline{B} = 0$	$B = 1$
$\overline{A} = 0$	0	1
$A = 1$	1	1

K-map سے ہم مندرجہ ذیل گروپس بنا سکتے ہیں:

دوسرے کالم میں 2، ایک کا گروپ $\overline{A}.B + A.B$

$$\overline{A}.B + A.B = (\overline{A} + A).B = 1.B = B$$

دوسری قطار میں 2، ایک کا گروپ $A.B + A.\overline{B}$

$$A.\overline{B} + A.B = (\overline{B} + B).A = 1.A = A$$

حتمی ایکسپریشن:

$$F(A, B) = B + A$$

عملی استعمال:

کرناف میپ ڈیجیٹل سرکٹ ڈیزائن میں بڑے پیمانے پر استعمال کیے جاتے ہیں تاکہ کسی مخصوص فنکشن کے لیے اہم گیٹس کی تعداد کو کم سے کم کیا جا سکے۔ اس سے ایسے سرکٹ بنتے ہیں جو تیز، سستے اور کم بجلی استعمال کرتے ہیں۔

سرگرمی

ایک ڈیجیٹل سرکٹ بنائیں جس میں دو 4 بٹ بائنری نمبر شامل کرنے کے لیے ہاف ایڈرز اور فل ایڈرز دونوں شامل ہوں۔ ہر قدم کے لیے ٹروتھ ٹیبل، بولین ایکسپریشن اور سرکٹ ڈایاگرام بنائیں۔

خلاصہ

- ڈیجیٹل سسٹم موجودہ دور کے الیکٹرانکس اور کمپیوٹنگ کی بنیاد ہیں۔ وہ ڈیجیٹل ڈیٹا چلاتے ہیں جو '1' اور '0' کی شکل میں ہے۔
- اینالاگ سگنلز کی نوعیت ایسی ہے کہ یہ مسلسل وقت کے ساتھ بدلنے والا سگنل ہے۔
- ADC اینالاگ سگنلز کو ڈیجیٹل سگنلز میں تبدیل کرنے کا عمل ہے جسے ڈیجیٹل آلات میں استعمال کیا جا سکتا ہے۔ مثال کے طور پر کمپیوٹر اور اسمارٹ فون۔
- DAC کا مطلب ڈیجیٹل سے اینالاگ کنورٹر ہے، یہ ڈیجیٹل سگنل کو واپس اینالاگ سگنل میں تبدیل کرنے کا طریقہ ہے۔
- ڈیجیٹل لاجک تمام ڈیجیٹل سسٹمز کی بنیاد ہے اور یہ وہ تکنیک ہے جو ہم ڈیجیٹل معلومات پر عمل کرنے کے لیے استعمال کرتے ہیں جو بائنری نمبروں کی شکل میں ہے۔
- بولین الجبرا ریاضی کا ایک ذیلی شعبہ ہے جو بائنری ویری ایبلز پر کارروائیوں پر مبنی ہے۔
- AND آپریشن کے معاملے میں آؤٹ پٹ صرف اس صورت میں 1 ہوتا ہے جب دونوں ان پٹ قیمتیں 1 ہوتی ہیں۔ بصورت دیگر آؤٹ پٹ فالز ہے (0)۔
- OR گیٹ کی فعالیت یہ ہے کہ اس کا نتیجہ '0' صرف اس وقت ہوتا ہے جب دونوں ان پٹ قیمتیں 0 ہوں۔ بصورت دیگر آؤٹ پٹ

ٹرو/ درست ہے(1)۔

- بولین الجبرا میں NOT آپریشن سب سے آسان لاجیکل کارروائیوں میں سے ایک ہے جو ایک بائنری ویری ایبل کو قبول کرتا ہے اور نتیجے کے طور پر ویری ایبل کی ویلیو الٹ کر دیتا ہے۔
- بولین فنکشن ریاضیاتی ایکسپریشنز ہیں جو بائنری ویرایبلز کے آپریشن کے سلسلے میں لاجک کی کارروائیوں کی موجودگی کی عکاسی کرتے ہیں۔
- تعمیر شدہ ڈیجیٹل سرکٹس کا ایک اہم عنصر لاجک ڈایاگرام ہے، جو لاجک گیٹس کے مابین رابطوں کی مدد سے سرکٹ کی ساخت کی نشاندہی کرتا ہے۔
- ایڈر سرکٹ بڑے پیمانے پر ڈیجیٹل الیکٹرانکس سسٹم میں ریاضی کے آپریشنز میں اہم ایپلی کیشن کے ساتھ استعمال ہوتے ہیں۔
- ہاف ایڈر ایک ڈیجیٹل سرکٹ ہے جو دو بائنری سنگل بٹ نمبروں کو جمع کرنے کے لیے استعمال ہوتا ہے۔
- ایک فل ایڈر ایک زیادہ پیچیدہ سرکٹ ہے جو تین سنگل بٹ بائنری نمبروں کو جمع کرتا ہے: دو اہم بٹس اور جمع کے لیے ایک کیری بٹ۔
- کرناف میپ ایک گرافک ایڈ ہے جو بولین اظہار کو آسان بنانے اور روزمرہ کے پیچیدہ الجبری آپریشنز کے بغیر لاجک فنکشن کو کم سے کم کرنے میں استعمال ہوتا ہے۔
- بولین الجبرا میں منٹرم ایک مخصوص فنکشن کے لیے بولین ایکسپریشن کی ایک مخصوص پروڈکٹ (AND) شکل ہے جس میں فنکشن کے ہر ویری ایبل کو عام یا مکمل شکل میں شامل کیا جاتا ہے۔

مشق

1۔ درست جواب کا انتخاب کریں:

(i) درج ذیل میں سے کون سا بولین ایکسپریشن OR آپریشن کی نمائندگی کرتا ہے؟

(الف) A.B (ب) B+A (ج) A (د) A+B

(ii) بولین ایکسپریشن A.0 = 0 کا مترادف ہے:

(الف) A + 1 = 1 (ب) A + 0 = A

(ج) A .1 = A (د) A.0=0

(iii) کون سا لاجک گیٹ صرف اس صورت میں درست جواب دیتا ہے جب دونوں ان پٹ درست ہوں؟

(الف) OR گیٹ (ب) AND گیٹ

(ج) XOR گیٹ (د) NOT گیٹ

(iv) ہاف ایڈر سرکٹ میں، کیری کس آپریشن کے ذریعے پیدا ہوتی ہے؟

(الف) XOR آپریشن (ب) AND آپریشن

(ج) OR آپریشن (د) NOT آپریشن

(v) بائنری نمبر 1101 کا ڈیسمل کیا ہے؟

(الف) 11 (ب) 12

(ج) 13 (د) 14

2۔ مختصر سوالات۔

(i) بولین فنکشن کی وضاحت کریں اور ایک مثال دیں۔

(ii) ڈیجیٹل لاجک میں ٹروتھ ٹیبل کی کیا اہمیت ہے؟

(iii) اینالاگ اور ڈیجیٹل سگنلز کے درمیان فرق کی وضاحت کریں۔

(iv) NOT گیٹ کے کام کو اس کے ٹروتھ ٹیبل کے ساتھ بیان کریں۔

(v) بولین ایکسپریشن کو آسان بنانے میں کارناف میپ کا مقصد کیا ہے؟

3۔ تفصیلی سوالات

(i) کمپیوٹرز میں بولین فنکشن کا استعمال لکھیں۔

(ii) بولین ایکسپریشن کے لیے ایک مثال کے ساتھ ٹروتھ ٹیبل کی تعمیر کی وضاحت کریں۔

(iii) بولین الجبرا میں دوہرے پن کے تصور کی وضاحت کریں اور اس کی وضاحت کرنے کے لیے ایک مثال فراہم کریں۔

(iv) ہاف ایڈرز اور فل ایڈرز کا موازنہ کریں بشمول ان کی ٹروتھ ٹیبل، بولین ایکسپریشن اور سرکٹ ڈایاگرام۔

(v) کارناف میپ بولین ایکسپریشن کو کیسے آسان بناتے ہیں؟ اقدامات کے ساتھ ایک تفصیلی مثال فراہم کریں۔

(vi) ہاف ایڈرز اور فل ایڈرز دونوں کا استعمال کرتے ہوئے 4 بٹ بائنری ایڈر ڈیزائن کریں۔ ٹروتھ ٹیبلوں، بولین ایکسپریشنز اور سرکٹ ڈایاگرام کے ساتھ ہر مرحلے کی وضاحت کریں۔

(vii) بولین الجبرا کے قواعد کا استعمال کرتے ہوئے مندرجہ ذیل بولین فنکشن کو آسان بنائیں:

$$F(A, B) = A \cdot B + A \cdot \overline{B}$$

(viii) درج ذیل فنکشن کو آسان بنانے کے لیے ڈی مورگن کے قوانین کا استعمال کریں:

$$F(A, B, C) = \overline{A + B} + \overline{AC}$$

(ix) مندرجہ ذیل ایکسپریشن کو آسان بنائیں:

(الف) $\overline{A + B} \cdot (A + B)$

(ب) $(A + \overline{B}) \cdot (\overline{A} + B)$

(ج) $A + \overline{A} \cdot (\overline{B} + C)$

(د) $\overline{A \cdot B} + A \cdot B$

(ہ) $(A \cdot B) + (\overline{A} \cdot \overline{B})$