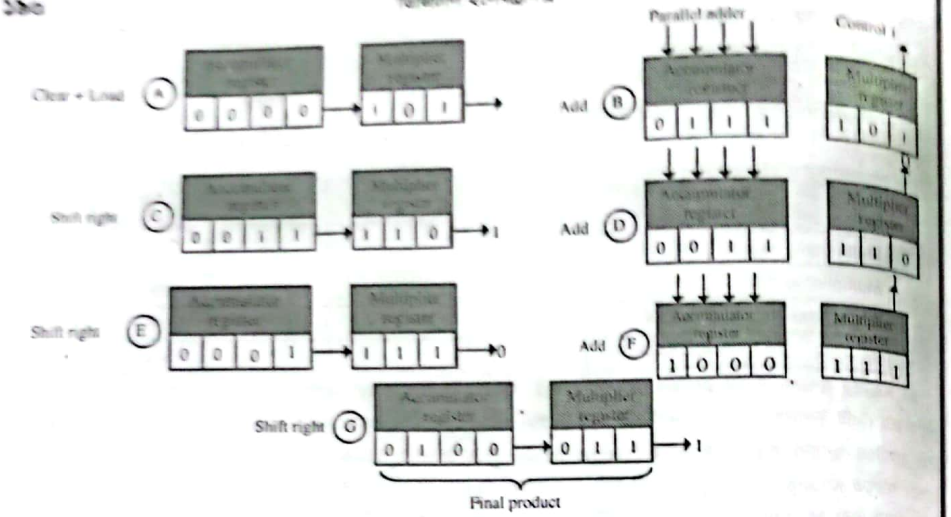


চিত্র ঃ ৯.১৫ (b)

এখানে বাইনারি $111(7_{10})$ কে 101 (5_{10}) দ্বারা গুণন প্রক্রিয়া দেখানো হয়েছে। এই পদ্ধতির তিনটি গুরুত্বপূর্ণ বিষয় নিয়ে আলোচনা করা হলো–

১। আংশিক গুণফল শূন্য হবে (000) যদি Multiplier 0 (শূন্য) হয় এবং Multiplicand এর সমান হবে, যদি Multiplier 1 (এক) হয়।

২। Product register অর্থাৎ যে রেজিস্টারে গুণফলের মান রাখা হবে, তার বিট সংখ্যা Multiplicand রেজিস্টারের বিট সংখ্যার দ্বিগুণ হবে।

৩। যোগের সময় আংশিক গুণফলটি ডানদিকে এক ঘর শিফট হবে।

এই ব্লক চিত্রটিতে বামদিকের উপরেরটি Multiplicand রেজিস্টার, মাঝে আছে 4-bit parallel Adder নিচের ডানদিকে অ্যাকুমুলেটর রেজিস্টার এবং Multiplier রেজিস্টার দেখানো হয়েছে। এখানে Accumulator এবং Multiplier রেজিস্টারকে একসঙ্গে বিবেচনা করতে হবে।

Line 1			1	1	1	Multiplicand
Line 2		×	1	0	1	Multiplier
Line 3			1	1	1	First partial product
Line 4		0	0	0		2nd partial product
Line 5		0	1	1	1	Temporary product (Line 3 + line 4)
Line 6	1	1	1			Third partial product
	1	0	0	0	1	1 Product

এখন আমরা বর্তনীটির কার্যপ্রণালি বিস্তারিতভাবে আলোচনা করব। উপরের বামদিকের Multiplicand রেজিস্টারে 111 লোড করা আছে। Accumulator-এ 0000 এবং Multiplier রেজিস্টারে 101 লোড করা আছে।

এখন Step-B এর (যা চিত্রে দেখানো হয়েছে) অবস্থা বর্ণনা করা হচ্ছে এবং যেখানে Control line- এ 1 প্রয়োগ করার সাথে সাথে 0000 এবং 111 যোগ হয়ে A রেজিস্টার 0111 হবে এবং Multiplier-এ 101 হবে।

Step-C তে Accumulator এবং Multiplier এর ডাটা ১ ঘর ডানে শিফট হবে এবং Multiplier এর LSB bit হারিয়ে যাবে অর্থাৎ 1টি বাহিরে চলে যাবে।

Step-D তে আবার যোগ কার্য সম্পন্ন Step হবে তবে এই পর্যায়ে Control line- এ 0 (শূন্য) প্রয়োগ হওয়ায় কোন যোগ কার্য সম্পন্ন হবে না, ফলে রেজিস্টারের Content যা ছিল তাই থাকবে।

Step-E তে আবার ডানদিকে ১ ঘর শিফট কার্য সম্পন্ন হবে। Accumulator রেজিস্টারের Content হবে 0001 এবং Multiplier Register এর Content হবে 111.

Step-F তে যোগ কার্য সম্পন্ন হয়ে 0001 + 111 = 1000 হবে, যা Accumulator- এ জমা থাকবে এবং Multiplier Register এর Content 111 হবে।

Step-G তে এক (১) ঘর ডানে শিফট হলে সম্পূর্ণ ডাটাটি হবে 0100011; এটি চূড়ান্ত গুণফল হিসাবে গণ্য হবে।

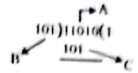
■ ডিভিসরের কার্যপ্রণালি (Operation of a divisor) ঃ

Binary division হলো Arithmetic operation এর একটি গুরুত্বপূর্ণ অংশ যা বাদ দেওয়া সম্ভব নয়। এটি binary arithmatic operation এর সবচেয়ে জটিল অংশ। যদিও এটি খুব কঠিন নয়, অন্য বাইনারি অপারেশনের তুলনায় একটু কঠিন। অন্য অপারেশনগুলোর নির্দিষ্ট কিছু নিয়ম আছে কিন্তু binary division operation এর নির্দিষ্ট কোন নিয়ম নেই। যদিও এই পদ্ধতিটি দশমিক বিভাজনের মতোই। নিচের একটি উদাহরণ দিয়ে একটি দেখানো হলো–

ধরি A = 11010 এবং B = 101

এখানে A কে B দ্বারা ভাগ করতে হবে,

অপারেশচের প্রক্রিয়াটি ধাপে ধাপে দেখলে এটি স্পষ্ট হবে।



A = Dividend
B = Divisor

আমরা যদি 101 কে 1 এর সাথে বাইনারি গুণ করি, তাহলে পাই $1 \times 1 = 1, 1 \times 0 = 0$ এবং $1 \times 1 = 1$ যা c দ্বারা দেখানো হয়েছে। তখন 110 হতে 101 কে সাবট্রাক্ট করলে 0011 বাইনারি সংখ্যাটি পাওয়া যায় যা নিচে দেখানো হলো–

$$\begin{array}{r} 101\overline{)11010(1} \\ \underline{101} \\ 0011 \end{array}$$

বিভাজনের নিয়ম অনুসারে LSB (Least Significant Bit) টি নেমে আসে এবং আমরা বিভাজক B এর সাথে পুনরায় ১ গুণ করতে চাই কিন্তু দেখা যায় $101 \times 1 = 101$ এই মানটি 0011 এর চেয়ে বেশি হয় তাই আমরা এই ধাপ অনুসরণ না করে পরবর্তিত ধাপ অনুসরণ করব।

এই ধাপে আমরা B = 101 এর সাথে 0 গুণ করে পুনরায় LSB কে ডান দিকে হতে নিচে নিয়ে আসব যা নিচে দেখানো হলো–

$$\begin{array}{r} 101\overline{)11010(10} \\ \underline{101} \\ 00110 \end{array}$$

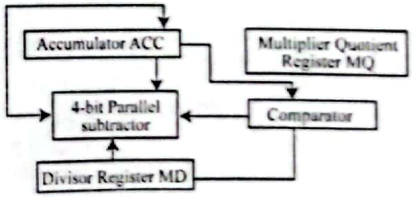
এখন আবার বিভাজক 1 দ্বারা গুণ করতে হবে। এখন দেখা যায় 00110 এর চেয়ে 101 ছোট তাই এই ধাপ অনুসরণ করব

$$\begin{array}{r} \text{ভাগফল} \\ 101\overline{)11010(101} \leftarrow \text{(quotient)} \\ \underline{101} \\ 00110 \\ \underline{101} \\ 001 \leftarrow \text{(remainder)} \end{array} \quad \text{ভাগশেষ}$$

এটিই বাইনারি ভাগের চূড়ান্ত ধাপ যেখানে ফলাফল হলো, ভাগফল = 101 এবং ভাগশেষ = 001

4-বিট সিকুয়েনশিয়াল বাইনারি ডিভাইডার (4-bit sequential binary divider) ঃ

৯.১৬ নং চিত্রে একটি বাইনারি সিকুয়েনশিয়াল ডিভাইডার সার্কিট দেখানো হয়েছে। এখান প্রথমেই ভাজ্য (Dividerd) কে অ্যাকুমুলেটরে (Acc) স্টোর করা হয় এবং ভাজক (Divisor) কে রেজিস্টার (MD) এ স্টোর করা হয়। ভাগকার্য সম্পাদনের পর ভাগফল Quotient Register (MQ) তে জমা হয়। প্রথমে MQ রেজিস্টার জিরোতে '0' তে সেট হয়।



চিত্র ঃ ৯.১৬ A 4-bit sequential binary divider

CamScanner

এবং ভাজ্যের সমতুল্য বাইনারি নাম্বার অ্যাকুমুলেটরে জমা হয় এবং ভাজকের সমতুল্য বাইনারি নাম্বার রিমিসিং রে (MD)-তে জমা হয়। MD রেজিস্টারে জমাকৃত ভাজকের মানকে অ্যাকুমুলেটরে অবস্থিত ভাজ্যের সাথে তুলনা করে কম্পারেটর MD ও ACC রেজিস্টারের ধারণকৃত ডাটাকে তুলনা করে। যদি ভাজক (Divisor) ছোট হয়, তবে এটি অবস্থিত ডাটা হতে বিয়োগ করবে এবং রেজিস্টারে LMB (Left Most Bit) এ একটি 0 প্রতিষ্ঠিত হবে। এই প্রসেসটি চলবে যতক্ষণ না ভাজ্যের (Dividend) সকল বিট ব্যবহৃত না হয় এবং সবগুলো বিট MQ রেজিস্টারে উৎপন্ন না হয়। এ একটি ভাগ কার্য সম্পাদন হয়।

উদাহরণ-৬। নিচের বাইনারি ভাগগুলো সম্পন্ন কর।

(i) 11001 ÷ 101

সমাধান:

```
101)11001(101
    101
     101
     101
```

(ii) 11101.01 ÷ 1100

সমাধান:

```
1100)11101.01(10.0111
     1100
     10101
      1100
      0010
       1100
       1100
```

∴ ভাগফল = 10.0111

(iii) 10110.1 ÷ 1101

সমাধান:

```
1101)10110.1(1.101
     1101
     10011
      1101
      11000
       1101
       1101
```

∴ ভাগফল = 1.101 এবং ভাগশেষ = 1011

(iv) 101.11 ÷ 111

সমাধান:

```
111)101.11(0.11
    111
    1001
     111
     10
```

∴ ভাগফল = 0.10 এবং ভাগশেষ = 10

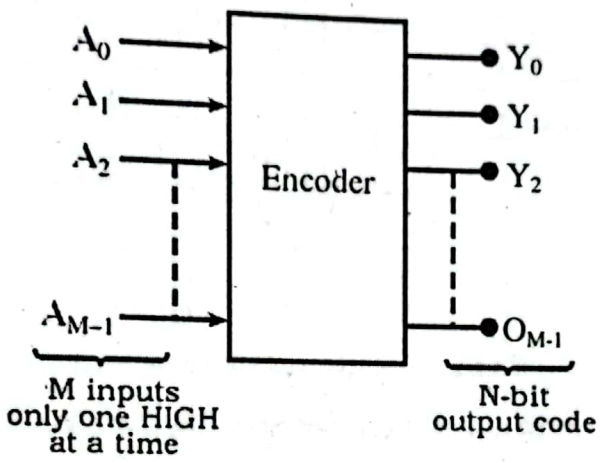
CamScanner

অধ্যায়-১০
এনকোডার
(Encoder)

১০.১ এনকোডার (Encoder) ঃ

❐ এনকোডার (Encoder) ঃ

এনকোডার এক ধরনের লজিক্যাল সার্কিট, যা ডেসিমেল সংখ্যার সমতুল্য বাইনারি সংখ্যা তৈরি করে। এই সার্কিটে সব সময় n সংখ্যক Input থাকবে এবং n সংখ্যক Output থাকবে। নিচে একটি এনকোডারের ব্লক চিত্র দেখানো হয়েছে–



চিত্র ঃ ১০.১ সাধারণ এনকোডার এর ডায়াগ্রাম

যে সার্কিটের মাধ্যমে অ্যানালগ ডাটাকে বাইনারিতে রূপান্তর করা যায় বা মেশিনের বোধগম্য ভাষায় রূপান্তর করা যায়, তাকে এনকোডার বলে। নিম্নে এনকোডারের ট্রুথ টেবিল ও সার্কিট ডায়াগ্রাম দেখানো হলো–

Input										Output			
A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	Y_0	Y_1	Y_2	Y_3
1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	0	0	0	0	1	0	1
0	0	0	0	0	0	1	0	0	0	0	1	1	0
0	0	0	0	0	0	0	1	0	0	0	1	1	1
0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0	0	1	1	0	0	1

ছক-১ ঃ এনকোডার ট্রুথ টেবিল