

الف)

①

Single Precision, 32 bit

1 bit sign 8 bit exponent 23 bit fraction

$$5.5 \left\{ \begin{array}{l} (5)_1 \Rightarrow (101)_2 \\ (0.5)_1 \Rightarrow (0.1)_2 \end{array} \right\} \Rightarrow (5.5)_1 \Rightarrow (101.1)_2$$

$$101.1 = 1.011 \times 2^2 \left\{ \begin{array}{l} \text{biased exponent } 127+2=129 \Rightarrow 10000001 \\ \text{normalised mantissa } 011 \end{array} \right.$$

Sign

$$\begin{array}{c} \uparrow \\ 0 \end{array} \underbrace{10000001}_{\text{Exponent}} \underbrace{01100000000000000000000}_{\text{mantissa}}$$

ب) نه سوال قبل عمل می کنیم چون از $Q_{2.5}$ استفاده می کنیم $\text{Scale} = \text{factor}$ می باشد

$$1.25 \times 10^{-45} \times 10^{45} = 1.25 \Rightarrow (1.01)_2$$

$$-6.25 \times 10^{-48} \times 10^{45} = -0.625 \Rightarrow -(0.101)_2$$

Sign

$$Q_{2.5} \Rightarrow 1.01 \Rightarrow \begin{array}{c} \text{Sign} \\ 0 \end{array} 01.01000$$

$$Q_{2.5} \Rightarrow -0.101 \Rightarrow \begin{array}{c} \text{Sign} \\ 1 \end{array} 00.10100 \text{ 2's complement}$$

Sign

$$11.01100$$

$$\begin{array}{r} \text{Add} \rightarrow 01.01000 \\ + 11.01100 \\ \hline 000.10100 \rightarrow 0.625 \end{array}$$

$$\frac{0.625}{1.45} \rightarrow 0.625 \times 10^{-45} \quad \text{بر Scaling factor تعیین می کنیم.}$$

(2)

re timing

این روش به منظور کاهش کمرنگی (critical path) و افزایش سرعت مدار سرعت می پذیرد و این کار با انتقال رجیسترهای pipeline به یک موقعیت بهینه انجام می دهد. re timing خودکار سازی این روش به شیوه های سیستماتیک با الگوریتم های مختلف می تواند به طور کلی که 2 روش زیر رعایت شوند:

- 1) اطمینان دادن از اینکه تابع های انتقال I/O مدار تغییر نمی کنند.
- 2) قابل اجرا در ابزار CAD باشند.

برای re timing سیستماتیک مدار دیجیتال تبدیل به یک data flow graph می شود و سپس با استفاده از تکنیک های گراف، رجیسترها به صورت سیستماتیک و بدون تغییر تابع انتقال I/O مربوط به data flow graph اصلی در طول combinational logic حرکت می کنند.

re pipelining: به طراحی های feed forward، این تکنیک رجیسترها را به ورودی و خروجی اضافه می کند و سپس این رجیسترها را با استفاده از تکنیک re timing در طول طراحی جایی می کند که بهترین بازدهی را بگیرد. این روش تعدادی رجیستر به pipeline اضافه می کند که این باعث افتادن شدن یک clock latency ثابت بین ورودی و خروجی می شود. هر چند که سایر خواص طراحی حفظ می شود.

C-Slow Retiming: شامل گذار C باره‌ی تمامی رجیسترها گراچی synchronous

است و بدون منتظر رجیسترها را حرکت می‌دهد. مدار را به C همت مطابقتی تقسیم می‌کند و بین ورودی‌ها و خروجی‌ها سوئیچ می‌شود.

Cut-Set Retiming: اجازه می‌دهد که گذار دلخواهی رجیسترها در یک مسیر رد به جوامع اضافه کنیم یا رجیسترها را از یک ورودی به خروجی یا برعکس منتقل کنیم و این در حالی است که تابع انتقال I/O ثابت می‌ماند و حفظ می‌شود.