

به نام خدا

## تکلیف معماری کامپیوتر سری ۳

پرهام الوانی

۴ اردیبهشت ۱۳۹۴

## فهرست مطالب

۲	۱ طراحی ۱
۲	۲ طراحی ۲
۳	۳ طراحی ۳

## ۱ طراحی ۱

در این طراحی تاخیر با فرض اینکه تاخیر adder ripple برابر با  $2d$  و تاخیر Multiplexer برابر با  $3d$  باشد داریم:

$$FA-1: 1 * 2d = 2d$$

$$\Rightarrow \text{Level}-1: 2d$$

$$FA-2: 1 * 2d = 2d$$

$$\text{Mux}-1: 3d$$

$$\Rightarrow \text{Level}-2: 5d$$

$$FA-3: 2 * 2d = 4d$$

$$\text{Mux}-2: 3d$$

$$\Rightarrow \text{Level}-3: 8d$$

$$FA-4: 4 * 2d = 8d$$

$$\text{Mux}-3: 3d$$

$$\Rightarrow \text{Level}-4: 11d$$

$$FA-5: 5 * 2d = 10d$$

$$\text{Mux}-4: 3d$$

$$\Rightarrow \text{Level}-5: 14d$$

$$FA-6: 3 * 2d = 6d$$

$$\Rightarrow \text{Level}-6: 17d$$

با توجه به آنچه در بالا گفته شد برای این طراحی نیاز به ۳۲ عدد Adder Full و ۱۰ عدد Multiplexer است.

## ۲ طراحی ۲

با توجه به اینکه در این طراحی از روابط SOP استفاده شده است پس تاخیر پس از محاسبه ی  $p_i$  ها و  $g_i$  ها برابر با  $2d$  خواهد بود. توجه به این نکته هم خالی از لطف نیست که برای پیاده سازی OR با ورودی های زیاد ممکن است نیاز شود سطح های پیاده سازی افزایش یافته و تاخیر بیشتر شود.

$$delay = 3d$$

و در نهایت برای تعداد قطعات مصرف شده در پیاده سازی انجام شده توسط نرم افزار proteus داریم:

Quantity	Name Part
۱۵	AND
۹	OR
۱۶	XOR
۶	AND-۳
۱	OR-۳
۱	OR-۴
۵	AND-۴
۴	AND-۵
۱	OR-۵
۵	AND-۷
۱	OR-۶
۱	OR-۷
۱	AND-۸
۱	OR-۸

### ۳ طراحی ۳

در این طراحی نیاز به ۲ عدد Adder Ripple ۸ بیتی داریم زیرا اگر حاصل جمع carry داشته باشد باید حاصل را با ۱ جمع کنیم. برای پیاده سازی بهینه تر Adder Ripple دوم را با استفاده از Adder Half ها پیاده سازی میکنیم.

$$Half - Adderdelay = ۱d$$

$$Full - Adderdelay = ۲d$$

$$Full - AdderbasedRipple - Adderdelay = ۲nd$$

$$\Rightarrow ۸ * ۲d = ۱۶d$$

$$Half - AdderbasedRipple - Adderdelay = nd$$

$$\Rightarrow ۸ * d = ۸d$$

$$\Rightarrow ۸d + ۱۶d = ۲۴d$$

با توجه آنچه در بالا گفته شد برای این طراحی نیاز به ۸ عدد Full-Adder و ۸ عدد Half-Adder است.