

# بهینه سازی

# معیارهای بهینه‌سازی

## • هدف:

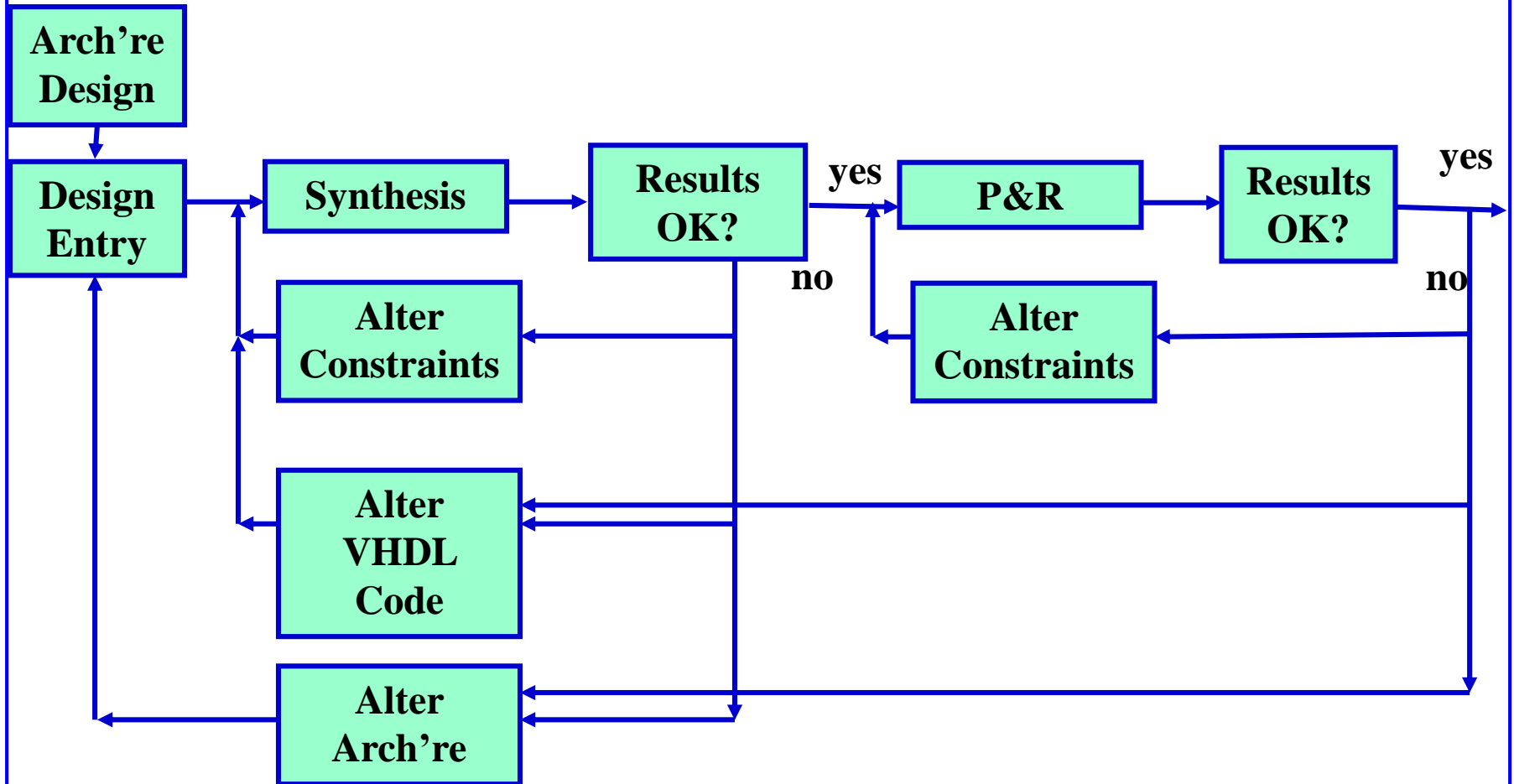
### □ طراحی بهتر:

- مساحت
- کارایی (سرعت)
- توان مصرفی

### □ تفاوت طراح خوب با طراح مبتدی:

- کد نویسی مناسب
- تعیین محدودیت‌های مناسب
- استفاده از سایر امکانات ابزار

# فرایند بهینه سازی



# بهینه‌سازی مساحت

## • بهینه سازی مساحت:

☐ کاهش یا متعادل کردن منابع مصرفی

☐ با هدف استفاده از تراشه ارزان تر

☐ معمولاً همراه با کاهش توان

- بر خلاف سرعت

# فنون بهینه‌سازی مساحت

## • تخمین مساحت:

□ برای تصمیم مناسب در هر مرحله

□ برای انتخاب تراشه در مراحل اولیه

← شروع زودتر طراحی مورد

□ مرحله قبل از سنتز:

- تجربی

- از طرح های مشابه گذشته

- اطلاعات IP

# فنون بهینه سازی مساحت

## • فنون بهینه سازی مساحت:

□ آشنایی با رفتار ابزار

← - کدنویسی مناسب

□ آشنایی با معماری تراشه

- بلوک های منطقی

- بلوک های محاسباتی

- حافظه

- ...

- مثال: ابزار اجازه می دهد محتویات LUT را دستی پر کنید

# فنون بهینه سازی مساحت

## • فنون بهینه سازی مساحت:

□ بررسی گزارش ابزارها

- مشاهده مصرف هر پودمان

- ← تمرکز روی بهینه سازی پرمصرف

- فدا کردن معیاری برای بهبود معیار دیگر ← طراح جلوگیری کند

- مثال: تکرار بخشی از مدار برای بهبود سرعت

# فنون بهینه سازی مساحت

## • فنون بهینه سازی مساحت:

□ اشتراک منابع (resource sharing)

- استفاده از منابع سخت افزاری مشترک برای
  - اعمال یکسان
  - اعمال مشابه (مانند جمع کننده، تفریق کننده، و مقایسه کننده) ولی مستقل
  - گاهی باعث کاهش فرکانس
  - می توان جلوگیری کرد

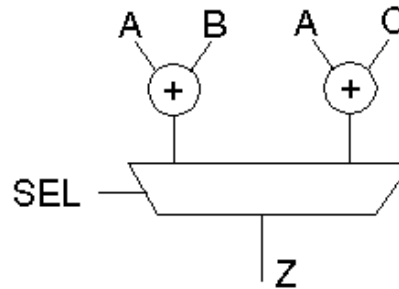


# فنون بهینه سازی مساحت

## • فنون بهینه سازی مساحت:

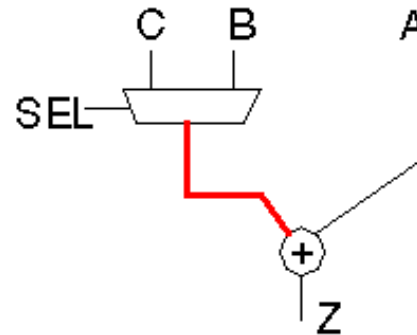
```
process (SEL,A,B)
begin
  if SEL = `1` then
    Z <= A + B;
  else
    Z <= A + C;
  end if;
end process;
```

اشتراک منابع (resource sharing) □



سرعت؟ □

```
process (SEL,A,B)
  variable TEMP : std_logic_vector (31 downto 0);
begin
  if SEL = `1` then
    TEMP := B;
  else
    TEMP := C;
  end if;
  Z <= A + TEMP;
end process;
```



# فنون بهینه سازی مساحت

```
process (CUR_STATE, IN1, IN2)
begin
  case STATE is
    when S0 =>
      NEXT_STATE <= S1;
      ZOUT <= IN1;
    when S1 =>
      if (IN1 = IN2) then
        NEXT_STATE <= S0;
        ZOUT <= IN1 * IN2;
      else
        NEXT_STATE <= S1;
        ZOUT <= IN1 - IN2;
      end if;
    when S2 =>
      if (IN1 = "0001_1010_0000_1111") then
        NEXT_STATE <= S1;
        ZOUT <= IN1 * IN2;
      else
        NEXT_STATE <= S3;
        ZOUT <= IN1 - IN2;
      END IF;
      ...
    when others =>
      ...
  end case;
end process;
```

# فنون بهینه‌سازی مساحت

```
process (CUR_STATE, IN1, IN2)
    variable TEMP1 : std_logic_vector (15 downto 0);
    variable TEMP2 : std_logic_vector (15 downto 0);
begin
    TEMP1 := IN1 * IN2;
    TEMP2 := IN1 - IN2;
    case STATE is
        when S0 =>
            NEXT_STATE <= S1;
            ZOUT <= IN1;
        when S1 =>
            if (IN1 = IN2) then
                NEXT_STATE <= S0;
                ZOUT <= TEMP1;
            else
                NEXT_STATE <= S1;
                ZOUT <= TEMP2;
            end if;
        when S2 =>
            if (IN1 = "0001_1010_0000_1111") then
                NEXT_STATE <= S1;
                ZOUT <= TEMP1;
            else
                NEXT_STATE <= S3;
                ZOUT <= TEMP2;
            END IF;
        ...
        when others =>
            ...
    end case;
end process;
```

# فنون بهینه سازی مساحت

## • فنون بهینه سازی مساحت:

### □ اشتراک منابع (resource sharing)

- ابزارها در حالت کلی نمی دانند کدام برای طراح مناسب تر
- ← توصیه می کنند طراح در توصیف مشخص کند.
- گاهی اتوماتیک resource sharing می کنند.
- مثلاً با محدودیت بهینه سازی مساحت
- ← برای اطمینان، در کد مشخص کنید.

# فنون بهینه سازی مساحت

## • فنون بهینه سازی مساحت:

تنظیم گزینه های ابزارها 

```
run -opt_mode area/speed
```

- گاهی باعث کاهش سرعت

- می توان روی پودمان(های) مشخصی اعمال کرد.

- اشتراک منابع برای بلوک پودمان خاص

```
attribute resource_sharing: string;
```

```
attribute resource_sharing of entity_name: entity is  
"{yes|no}";
```

# فنون بهینه‌سازی مساحت

## • فنون بهینه‌سازی مساحت:

□ پیاده‌سازی مدارهای منطقی و محاسباتی و FSM با حافظه

- اگر حافظه اضافی داریم

- با کدنویسی مناسب

□ با ایجاد محدودیت:

```
attribute bram_map: string;
```

```
attribute bram_map of MY_COMPONENT : component is "yes";
```

```
attribute fsm_style: string;
```

```
attribute fsm style of MY ENTITY entity is "bram";
```

# فنون بهینه سازی مساحت

## • فنون بهینه سازی مساحت:

□ انتخاب مناسب کدهای حالت FSM

– سعی و خطا

□ استفاده از سیگنال های کنترلی یکسان برای ثبات های گوناگون

– Clock

– Reset

– Enable

– ← نگاشت این ثبات ها روی یک LB (مثلاً slice)

# فنون بهینه سازی مساحت

- فنون بهینه سازی مساحت:

□ مسطح سازی



# فنون بهینه‌سازی مساحت

## • فنون بهینه‌سازی مساحت:

□ استفاده متوازن از منابع سخت افزاری

LUT –

– حافظه بلوکی

– حافظه توزیع شده

– بلوک محاسباتی

– ...

□ بعضی چندکاره

– ← اجتناب از مصرف بی رویه یک نوع و بیکار ماندن نوع دیگر