

Synthesizing intARGB given component values

Assuming all the component values are legal

We are *given*

```
alpha == 0000 0000 0000 0000 0000 0000 AAAA AAAA
red    == 0000 0000 0000 0000 0000 0000 RRRR RRRR
green  == 0000 0000 0000 0000 0000 0000 GGGG GGGG
blue   == 0000 0000 0000 0000 0000 0000 BBBB BBBB
```

We want to *synthesize*

```
3          2          1
1          3          5          7          0
AAAA AAAA RRRR RRRR GGGG GGGG BBBB BBBB
```

We need to get the component's bits into the proper position

```
alpha * 2563 == alpha << 24 == AAAA AAAA 0000 0000 0000 0000 0000 0000
red    * 2562 == red    << 16 == 0000 0000 RRRR RRRR 0000 0000 0000 0000
green  * 256    == green <<  8 == 0000 0000 0000 0000 GGGG GGGG 0000 0000
blue           == blue           0000 0000 0000 0000 0000 0000 BBBB BBBB
```

And combine them into a single int value.

What if component values *outside* the legal range of [0,255]?

For example...

```
alpha == 255 == 0000 0000 0000 0000 0000 0000 1111 1111
red    == 257 == 0000 0000 0000 0000 0000 0001 0000 0001
green  == -1  == 1111 1111 1111 1111 1111 1111 1111 1111
blue   == 255 == 0000 0000 0000 0000 0000 0000 1111 1111
```

Then this is *not* going to give us the expected color...

```
1111 1111 0000 0000 0000 0000 0000 0000
0000 0001 0000 0001 0000 0000 0000 0000
1111 1111 1111 1111 1111 1111 0000 0000
0000 0000 0000 0000 0000 0000 1111 1111
```