

```

actor YCrCbtoRGB()
  int (size=10) Y, int (size=10) Cr, int (size=10) Cb =>
  int (size=8) R, int (size=8) G, int (size=8) B :

  int (size=13) rv = 292;
  int (size=13) gu = 101;
  int (size=13) gv = 149;
  int (size=13) bu = 520;
  int (size=11) t1 := 1023;

action
  Y:[y], Cr:[cr], Cb:[cb] =>
  R:[r], G:[g], B:[b]

var
  int (size=10) r, int (size=10) g, int (size=10) b, int (size=10) rt,
  int (size=10) gt, int (size=10) bt, int (size=11) yt, int (size=11) crt,
  int (size=11) cbt

do
  //signed to unsigned representation
  yt := bitand(y, t1);
  crt := bitand(cr, t1);
  cbt := bitand(cb, t1);
  //core algorithm
  rt := (((yt-64) << 8) + rv*(crt-512)) >> 10;
  gt := (((yt-64) << 8) - gu*(cbt-512) - gv*(crt-512)) >> 10;
  bt := (((yt-64) << 8) + bu*(cbt-512)) >> 10;
  //clip output r
  if (rt > 0) then
    if (rt < 255) then r := rt;
    else r := 255; end
  else
    r := 0;
  end
  //clip output g
  if (gt > 0) then
    if (gt < 255) then g := gt;
    else g := 255; end
  else
    g := 0;
  end
  //clip output b
  if (bt > 0) then
    if (bt < 255) then b := bt;
    else b := 255; end
  else
    b := 0;
  end
end
end

```