APPENDIX G

AHDL

G1. The behavioral AHDL code of the block DCELL

```
// Spectre AHDL for "TEZ", "DCELL", "ahdl"

module DCELL (Vinn, Vinp, Vbias, Icell, Vphi)
(gain,vth,power,cr,dvdt)
node [V , I] Vinn,Vinp,Icell,Vphi,Vbias;
parameter real gain=1 from [0:inf); //transconductance of transistor
parameter real vth=0.9 from [0:2];  // treshold voltage of
transistor
parameter real power=5 from [0:10]; // power supply
parameter real cr=0.296 from [0:1); // aspect ratio of input C
parameter real dvdt=0.11 from (0:inf); //change of stored voltage
with time at 100 degre
{     export real id1,id2;
      export real start_time;
      export real vb;
      export real vbakt;
      export real Tp,Tn;
      analog
      {
            if(V(Vphi)<=2.5)
            {     Tn=V(Vinn);
            Tp=V(Vinp);
                  start_time=$time();
            vbakt=V(Vbias);
                  vb=V(Vbias);      }
            else
            {     Tn=Tn;
            Tp=Tp;
                  start_time=start_time;
                  vbakt=vbakt;
            vb=vbakt+dvdt*($time()-start_time); }
            if ((vb+cr*(V(Vinp)-Tp))<power-vth)
            {id1=pow((power-vth-vb-cr*(V(Vinp)-Tp)),2);}
            else
            {id1=0;}
            if ((vb+cr*(V(Vinn)-Tn))<power-vth)
            {id2=pow((power-vth-vb-cr*(V(Vinn)-Tn)),2);}
            else
            {id2=0;}
            I(Icell)<-(-1)*gain*(id1+id2);
      }
 } // DCELL;
```

G2. The behavioral AHDL code of the block ROW_AMP

```
// Spectre AHDL for "TEZ", "Row_amp_ahdl", "ahdl"

module Row_amp_ahdl (in,out) (Ibias,Vref)
node [V , I] in, out;
parameter real Ibias=30u from [0:inf);
parameter real Vref=0.83 from [0:5];
{
analog
{
      V(in)<-Vref;
      I(out)<-Ibias-I(in);    }
} // Row_amp_ahdl;
```

G3. The behavioral AHDL code of the block ROW_WTA

```
// Spectre AHDL for "TEZ", "Row_WTA_ahdl", "ahdl"
module Row_WTA_ahdl (inb,outb) (Ibias,LH,LL)
node [V,I] inb[64],outb[64];
parameter real Ibias=30u from [0:inf);
parameter real LH=4.8 from [0:5];
parameter real LL=1.73 from [0:5];
{     export real Imax=0;
integer i;
analog
{
      generate(i=0:63:1)
            {if (i==0)
                  {Imax=Ibias-I(inb[i]);}
            else
                  {if ( Imax< (Ibias-I(inb[i])) )
                        {Imax=Ibias-I(inb[i]);}
                  else
                  {Imax=Imax;}}
            V(inb[i])<-0.8;}
      generate(i=0:63:1)
            {if ( Imax == Ibias-I(inb[i]) )
                  {V(outb[i])<-LL;   }
            else
                  {V(outb[i])<-LH; }}}
} // Row_WTA_ahdl;
```

G4. The behavioral AHDL code of the block DAC6

```
module DAC6(RN, RP, b0, b1, b2, b4, b5, b6, DACOUT,Vkont,VDDA,GNDA)
    node [V,I] RN;
    node [V,I] RP;
    node [V,I] b0;
    node [V,I] b1;
    node [V,I] b2;
    node [V,I] b4;
    node [V,I] b5;
    node [V,I] b6;
    node [V,I] DACOUT;
    node [V,I] Vkont;
    node [V,I] VDDA;
    node [V,I] GNDA;
{   node [V,I] DACOUT1;
      real out, code, vrn, vrp ;
      real rise = 20e-9;
      real fall = 20e-9;
      analog{
            code = 0;
            code += (V(b6) > 2.5) ?  32 : 0;
            code += (V(b5) > 2.5) ?  16 : 0;
            code += (V(b4) > 2.5) ?   8 : 0;
            code += (V(b2) > 2.5) ?   4 : 0;
            code += (V(b1) > 2.5) ?   2 : 0;
            code += (V(b0) > 2.5) ?   1 : 0;
            vrn = V(RN);
            vrp = V(RP);
            if( V(VDDA) > V(GNDA) )
                  {out = (vrn + ((vrp-vrn)/63)*code);}
            else
                  {out = 0.0;}
            V(DACOUT1) <- $transition( out, 0, rise, fall );
            if (V(Vkont) > 2.5 )
                  {V(DACOUT1,DACOUT) <- 0.0;}
            else
                  {I(DACOUT1,DACOUT) <- 0.0;} }
} // DAC6_ahdl;
```