Design-build of a heat transport system for electronic components

Spanidis, M.

ABSTRACT

Adequate cooling to electronic components is critical to optimize computer performance and prevent component failure. This paper presents a methodology for the preliminary thermal design of a heat transport system to cool a central processing unit (CPU). Preliminary design calculations have been implemented in the R programming language for computational convenience. These calculations were used to produced CAD of heat exchangers. CFD was performed on the primary heat exchange within the heat transport system. Results were used to provide an informed decision to purchase the components necessary to construct the system. Theoretical calculations were compared against the constructed system.

CONTENTS

NOMENCLATURE

T_j	Junction temperature	℃	n_f	Number of fins	-
T_{s1}	Surface temperature of the processor	℃	n_c	Number of flow channels	-
T_{s2}	Surface temperature of the heat sink	℃	A	Heat transfer area	m^2
T_w	Bulk water temperature within HX1	℃	R	Resistance due to heat transfer	$m\ ℃\ W^{-1}$
T_1	Inlet water temperature	℃	u	Velocity	$m\ s^{-1}$
T_2	Inlet water temperature	℃	d_i	Inner diameter	m
t_1	Inlet air temperature	℃	d_o	Outer diameter	m
t_2	Outlet air temperature	℃	Nu_{for}	Forced convection Nusselt number	-
Q	CPU thermal power	W	Nu	Nusselt number	-
l	Length	m	rad	Radiator	-
w	Width	m	w	Water	-
L	Height	m	a	Air	-
k	Thermal conductivity	$W\ m^{-1}\ ℃^{-1}$	man	Radiator manifold	-
h	Heat transfer coefficient	$W\ m^{-2}\ ℃^{-1}$	a	Processor	-
C_p	Heat capacity	$J\ kg^{-1}\ ℃^{-1}$	b	Heat sink	-
Re	Reynolds number	-	c	Channel	-
Pr	Prandtl number	-	f	Fanning friction factor	-
μ	Viscosity	Pa s	V	Volumetric flowrate	$m^3\ s^{-1}$
ρ	Density	$kg\ m^{-3}$	W	Shaft work	W

Introduction

Computer components such as the graphical processing unit and CPU can be damaged if temperatures exceed critical values. In some cases, these components are operated above manufacturer recommended settings to achieve increased performance in a process known as overclocking; this process significantly increases the temperature of the components and can lead to component failure should cooling to the components be unsatisfactory. Adequate cooling to these components is essential to optimize computer performance and prevent component failure.

Liquid and air cooling are the two primary methods used to cool these components. The former option is most popular amongst users that frequently exceed component nominal loads. The capital cost of a liquid cooling system significantly exceeds air cooled systems. This cost can vary significantly (by hundreds of dollars) depending on the number of components to cool, the heat transfer equipment used, and the pump required to deliver coolant.

The aim of this work is to use principles of heat transfer to develop the process and thermal design for a heat transport system (HTS) to cool a CPU. The design is such that the junction temperature does not exceed 50% of the maximum allowable temperature (50℃) at the processor die during steady-state conditions. The design has been used to provide an informed decision to purchase the components necessary to construct the system.

Process Design

The HTS requires a minimum of two heat exchanges. The first heat exchanger will provide relief to the power generated at the processor die and allow the junction temperature to be maintained at 50°C. The second heat exchanger will be used to maintain a constant coolant temperature.

Figure 1 illustrates the process flow. Coolant is charged to a pump reservoir such that there is satisfactory coolant for the HTS. A pump transfers the coolant to HX1, which relieves the load generated at the processor die to maintain a junction temperature of 50°C. The coolant at the outlet of HX1 continues throughout the loop to HX2, where it is air-cooled to the inlet temperature of HX1. Coolant continues through the loop back to the pump and reservoir and the cycle is continued. A drain valve is located before the pump reservoir should to system require drainage.

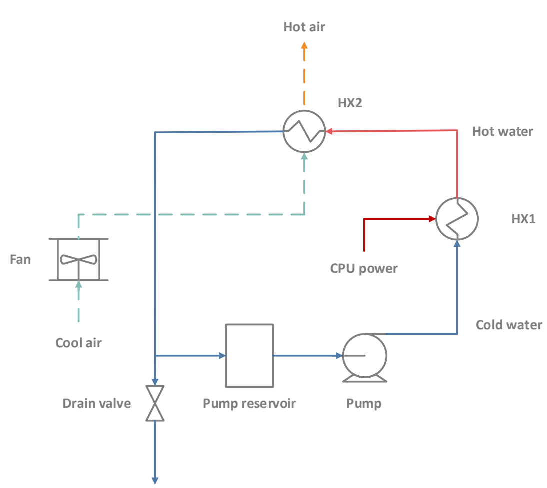


Figure 1
Process flow diagram.

Thermal Modelling

HX1

Figure 2 illustrates a general schematic for a water block (HX1). The water block assembly consists of a heat sink, a block casing, and inlets/outlets for coolant. The heat sink is made from thermally conductive material to draw heat from the CPU die.

The transfer of heat from the processor die to the coolant occurs as depicted in Figure 3. T_j is maintained through the convective heat transfer that occurs between the bulk coolant and the heat sink interface and the subsequent conductive heat transfer through the heat sink to the processor die.

Assuming that heat is transferred strictly in the x-direction, the heat sink and CPU surface temperatures, T_{s1} and T_{s2}, can be determined through Equation 1 and Equation 2 respectively.

$$T_{s1} = T_j - \frac{Q \times L_A}{A_A k_A} \tag{1}$$

$$T_{s2} = T_{s1} - \frac{Q \times L_B}{A_B k_B} \tag{2}$$

It is assumed that there is no thermal resistance at the coolant-heat sink interface, and that the heat transfer coefficient of the bulk coolant is constant throughout the water block. The heat transfer coefficient of the coolant can be evaluated using Equation 3.

$$h = \frac{h \times Nu}{D_h} \tag{3}$$

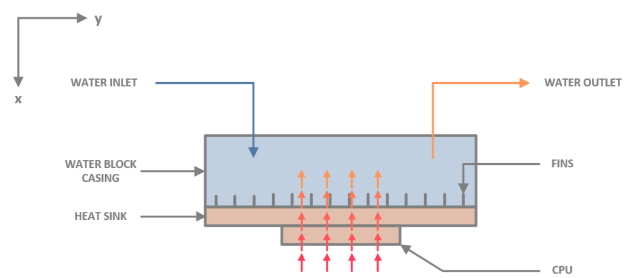


Figure 2
Water block schematic.

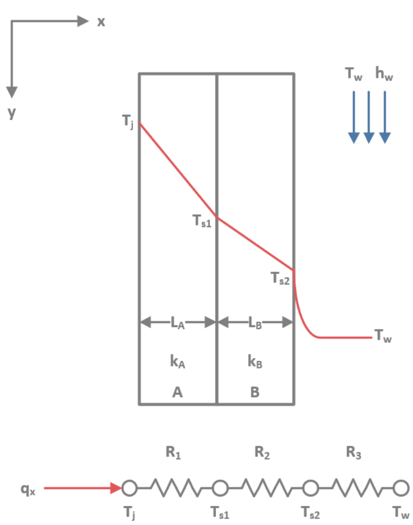


Figure 3
Heat transfer through the water block.

The Nusselt number can be evaluated using the general Nusselt number of heat exchangers servicing nonviscious liquids as expressed in Equation 4 [1]. Note that it is assumed that the viscosity of the fluid at the heat sink-fluid interface is equivalent to the viscosity of the bulk fluid.

$$Nu_w = 0.023 \times Re^{0.8} \times Pr^{0.33} \quad (4)$$

Reynolds and Prandtl numbers can be evaluated using Equation 5 and Equation 6 respectively. The hydraulic diameter can be approximated using the casing height and heat sink length.

$$Re = \frac{u \times \rho \times D_h}{\mu} \quad (5)$$

$$Pr = \frac{c_p \times \mu}{k} \quad (6)$$

Finally, the heat transfer area (and hence the heat sink geometry) can be approximated via Algorithm 1.

Algorithm 1

Approximating the heat sink heat transfer area to the bulk fluid.

input: $Q,\ l_B, w_B, h_f, w_f, n_f, T_w, T_{s2}$

while $Q_{calc} < Q$

$n_f += 1$

$A_{ht,wb} = l_B * w_B + 2 \times n_f \times l_B \times h_f$

$Q_{calc} = h \times A_{ht,wb} \times (T_w - T_{s2})$

end

HX2

Figure 4 illustrates a general schematic for a radiator (HX2). The radiator consists of an array of flow channels. Air is forced over these flow channels to return the coolant temperature the its desired temperature. Radiators are conventionally finned between flow channels; however, for simplicity of calculations fins have been ignored.

The transfer of heat from the coolant to the air occurs as depicted in Figure 5. $T_{1,HX2}$ is cooled to $T_{1,HX1}$ through forced convection of cool air across the flow channels. It is assumed that the temperature of the air and water at the surface of the channel is equal to the temperature of the bulk fluids.

Preliminary sizing of the radiator is conducted using Brown's method [2]. Equation 7 is evaluated using an assumed overall heat transfer coefficient to approximate the air temperature leaving the radiator.

$$t_2 = 0.050 \times U_{as} \times \left(\frac{T_1 + T_2}{2} - t_1\right) + t_1 \quad (7)$$

The heat transfer area of the radiator can be calculated by considered the number of flow channels and is using Equation 8.

$$A_{ht,rad} = 2 \times l_c \times w_c \times n_c + 2 \times l_c \times h_{rad} \times n_c \quad (8)$$

where

$$w_c = \frac{w_{rad}}{2 \times n_c - 1}$$

and

$$l_c = l_{rad} - 2 \times l_{man}$$

The hydraulic diameter of air flowing between the flow channels can be expressed as in Equation 9.

$$D_h = \frac{4 \times (l_{rad} \times w_{rad} - n_c \times l_c \times w_c)}{(n_c - 1) \times (2 \times w_c + l_c)} \quad (9)$$

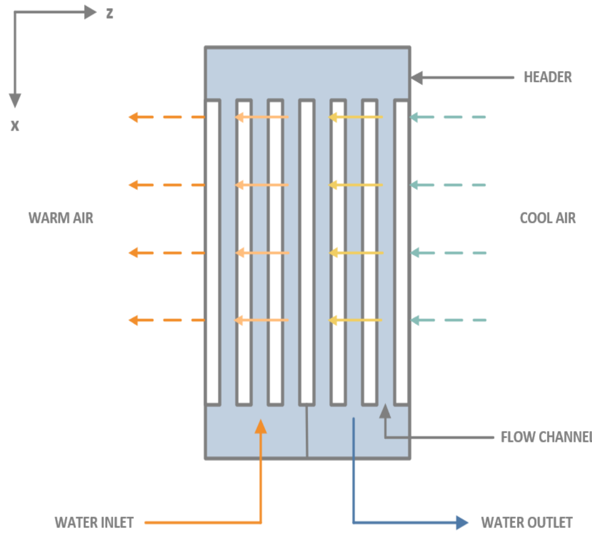


Figure 4

Radiator schematic.

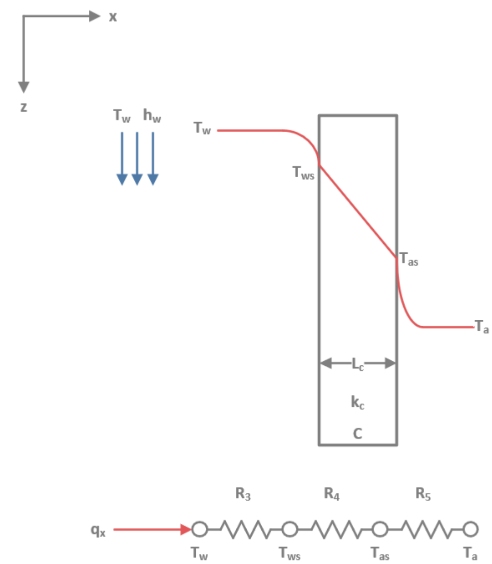


Figure 5

Heat transfer through the radiator.

Coolant velocity in the channel is evaluated via Equation 10.

$$u_{w,rad} = \frac{u_w \times \left(\left(\frac{\pi}{4}\right) \times (d_i)^2\right)}{w_c \times h_{rad}} \quad (10)$$

The heat transfer coefficient of the coolant in the radiator can be evaluated as described for *HX1*. The heat transfer coefficient of air is determined using the Nusselt number for forced convection as expressed in Equation 11.

$$Nu_{for} = 0.158 \times Re^{0.66} \times Pr^{0.37} \quad (11)$$

The velocity of air forced over the flow channels can be determined using Equation 12.

$$u_a = \frac{\dfrac{Q}{c_{p,a} \times (t_2 - t_1)} \Big/ \rho_a}{\pi \times \left(\left(\frac{l_{rad}}{2}\right)^2 - r_{motor}^2\right)} \quad (12)$$

Finally, the heat transfer area calculated using Equation 8 is verified against the heat transfer area calculated using the assumed heat transfer coefficient. Algorithm 2 describes the process to size the radiator.

Algorithm 2
Approximating the heat transfer area of the radiator.

input: $l_c, w_c, U_{as}, d_i, k_c, h_{rad}$
while $A_{as} < A_{calc}$
$n_c = n_c + 1$
for even n_c

$$A_{as} = \frac{Q}{U_{as} \times (t_2 - t_1)}$$

$$A_{ht,rad} = 2 \times l_c \times w_c * n_C + 2 \times l_c \times h_{rad} \times n_C$$

end

$$U_{calc} = \left(\frac{1}{h_w} + \frac{1}{h_a} + d_i \times \frac{\ln \frac{d_o}{d_i}}{2 \times k_c} \right)^{-1}$$

$U_{as} = U_{calc}$
end

Pumping requirements

Fiction factors are calculated using the Serghides analytical solution to the Darcy-Weisbach friction factor as expressed in Equation 13 [3].

$$\begin{aligned} A &= -2 \times log10\left(\frac{\epsilon}{3.7 \times d_i} + \frac{12}{Re}\right) \\ B &= -2 \times log10\left(\frac{\epsilon}{3.7 \times d_i} + 2.51 \times \frac{A}{Re}\right) \\ C &= -2 \times log10\left(\frac{\epsilon}{3.7 \times d_i} + 2.51 \times \frac{B}{Re}\right) \\ f &= \left(A - \left(\frac{(B-A)^2}{C - (2 \times B) + A}\right)\right)^{-2} \end{aligned} \tag{13}$$

Frictional losses associated with the pumping of coolant through tubing can be evaluated using Equation 14.

$$F_{tubing} = \frac{\left(8 * l_t * \left(\frac{2}{d_i}\right) + 2.5\right) * \frac{\rho_w * u_w^2}{2}}{\rho_w} \tag{14}$$

Losses through heat transfer equipment are approximated using twice the frictional losses that occur within the radiator; this is expressed in Equation 15.

$$F_{units} = 4 \times \left(8 \times f_{rad} \times \left(\frac{l_c}{d_i}\right) + 2.5\right) \times \frac{\rho_w * u_{w,rad}^2}{2} \times \left(\frac{u_w^2}{2 * 9.81}\right) \tag{15}$$

Frictional losses through sudden expansion and contractions within flow entrances and exits can be evaluated using the 2-K method described in [4]; these associated frictional losses are expressed in Equation 16.

$$F_{ee} = 3 \times \left(\left(\frac{160}{Re} + 0.50\right) + (1.0)\right) \times \left(\frac{u_w^2}{2 \times 9.81}\right) \tag{16}$$

Pumping power requirement can then be approximated following Equation 17.

$$W = -1 \times (F_{tubing} + F_{units} + F_{ee}) \times u_W \times \left(\left(\frac{\pi}{4}\right) * (d_i)^2\right) \times \rho \tag{17}$$

Methodology

Algorithm 1 and Algorithm 2 have been implemented in the R programming language. The script used has been appended to this document (all units are SI). These calculations have been used to generate preliminary sizing specifications for HX1 and HX2. All fluid properties are evaluated at inlet temperatures. The thermal power of the CPU, among other thermal and geometric properties of the CPU is available through [5]. It is assumed that the CPU is made of pure copper. Coolant temperature was assumed to be available at 25℃.

Preliminary sizing specifications were used to develop CAD models of equipment. CFD was preformed on the CAD of HX1 and compared to the preliminary calculations to evaluate the accuracy of the thermal design.

Equipment was purchased and assembled using the knowledge obtained through the preliminary thermal design and CFD. Comparisons against actual cooling performance across CFD and preliminary designs were made.

Results

Table 1 summarises the key parameters from the preliminary sizing of equipment; note that some parameters such as *Q*, *l*, etc. are pre-defined by the user as inputs for the algorithms (see Appendix 1).

Table 2 summarises some available specifications for equipment that was purchased based on the preliminary design and CFD.

Table 3 summarises steady state readings of *Tj* under nominal CPU load and maximum CPU load; if CPU thermal design power increases linearly with CPU operating frequency, this value should be 130 W.

Figure 6 illustrates the CAD of the HTS. Figure 7 and Figure 8 show the CAD of HX1 and HX2 respectively. Figure 9 provides various viewpoints of the CFD conducted on HX1; the CFD report has been appended to this document. Figure 10 shows a photo of the assembled HTS.

Table 1
Summary of preliminary sizing results.

Parameter	Value
HX1	
h_B	0.051
l_B	0.051
w_B	0.0063
T_2	25.5
n_f	12
l_f	0.05
w_f	0.001
HX2	
h_C	0.030
l_C	0.125
w_C	0.00051
t_2	25.5
n_c	162
Pump	
W	19.3
V	0.000032
Fan	
V	0.19

Table 2
Summary of purchased equipment specifications

Parameter	Value
HX1	
h_B	0.085
l_B	0.070
w_B	0.042
n_f	54
l_f	0.05
w_f	0.0006
HX2	
h_C	0.040
l_C	0.150
w_C	0.001
n_c	14
Pump	
W	37
Q	0.00005
Fan	
Q	0.019

Table 3
Real steady state temperature readings of T_j.

Parameter	Value
Nominal	30
Maximum load	50

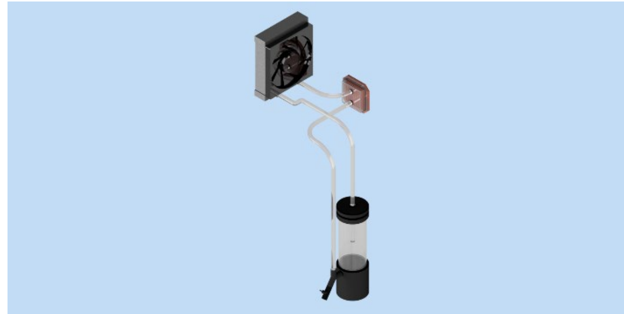


Figure 6
CAD of HTS.

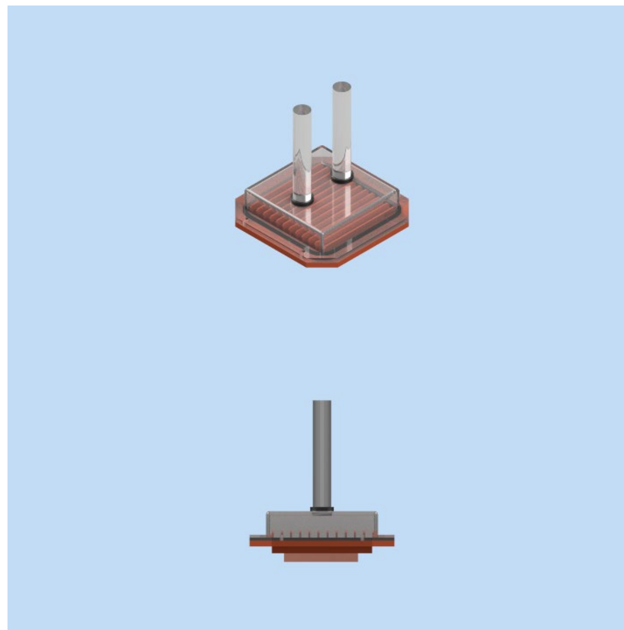


Figure 7
CAD of HX1.

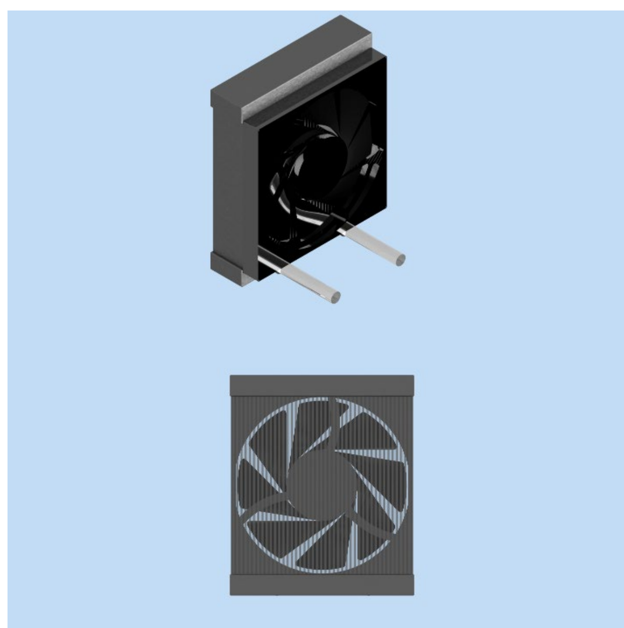


Figure 8
CAD of HX2.

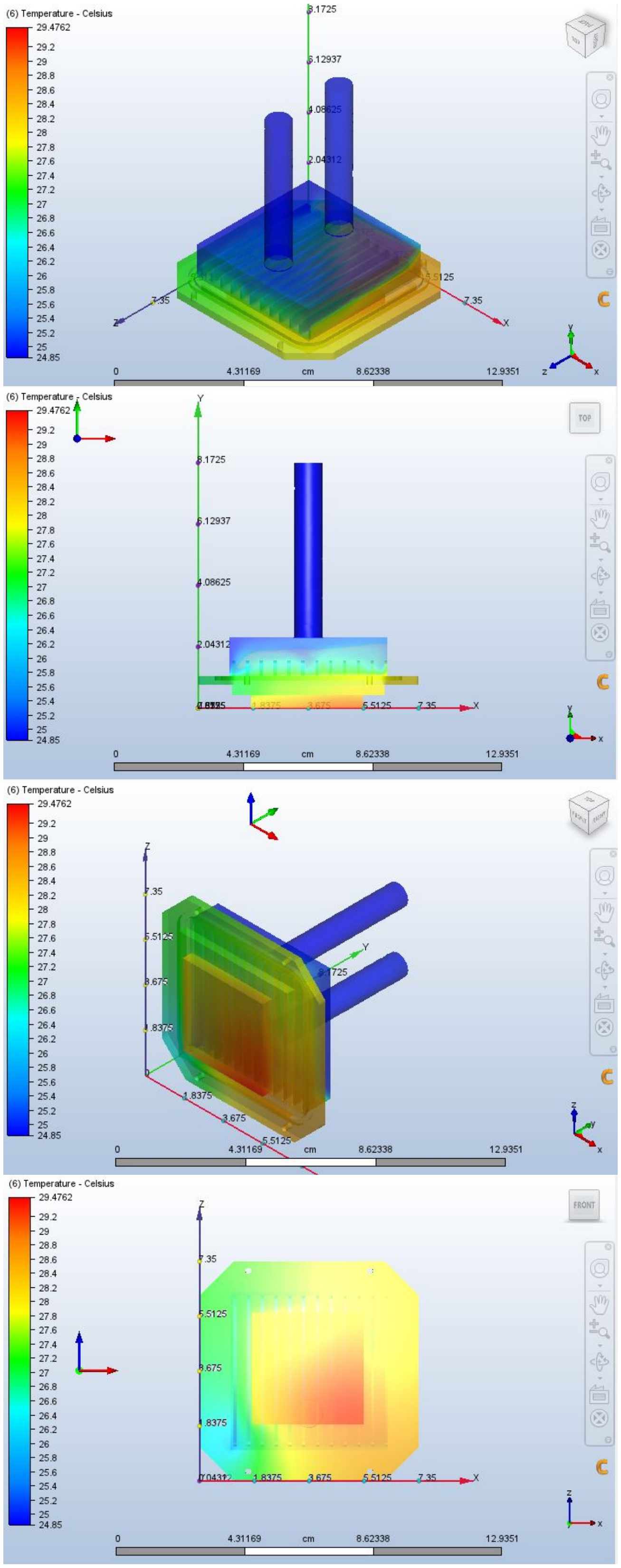


Figure 9
CFD of HX1.

Figure 10
Assembled HTS.

Discussion

It was unnecessary to preform CFD on HX2 as the coolant outlet temperature was virtually the same as the inlet temperature. CFD confirmed that the outlet temperature of HX1 is maintained at the inlet temperature. CFD on HX1 showed a steady state $Tj \approx 29℃$ which is in error of the Tj used in preliminary sizing by 42%. This is likely due to the increase in actual heat sink size of HX1 as additional material was used allow for the block casing, fastenings, and the o-ring. In addition, heat transfer for the CFD does not occur strictly in the x-axis but rather along all axis. It is plausible that both the directions of heat transfer and extra material used in the CAD of HX1 provided sufficient cooling to result in such error.

The number of channels required for cooling in HX2 is drastically different from the number of fluid channels traditionally found in radiators. Actual radiators come with finned surfaces between flow channels to promote heat transfer rather than additional flow channels. The preliminary sizing of the radiator, without considering the fins in the analysis, is unsatisfactory to provide a general understanding of radiator size requirements.

The number of flow channels also increased the pressure drop, as the hydraulic diameter of the flow channel is inversely proportional to the number of flow channels. It is likely that the pumping requirements calculated during preliminary sizing is in excess of the optimal value. Nevertheless, the preliminary pumping power required provided general insight as to the actual power needed. The calculations were preformed in a "worst-case-scenario" whereby the excessive fluid channels and the 2m assumption of tubing length (the actual tubing length is just over 1m) provided excessive frictional losses and hence more than necessary pumping power. Therefore, it was reasonable to assume that a pump that could provide more power than that calculated would be more than satisfactory.

Actual readings of Tj from the assembled system are provided in Table 3. The steady state Tj at nominal load is roughly 30℃ which aligns with CFD results; however, both the fluid velocity delivered to HX1 and the heat transfer area of HX1 vary from theoretical values (u_w varied by approximately +56%, A_{ht} by+ 437%). The lack of reduction in Tj given these flow velocity and heat transfer area increases is expected to be a result of the steady state coolant temperature delivered to the water block. As illustrated in Figure 11, the pump reservoir sits inside the computer casing, where it is exposed to heat from other electronic components (GPU, motherboard, etc.). It is expected that at steady state, the reservoir temperature increases to the steady state temperature of the case, which is likely to be approximately 30℃.

Conclusion

Preliminary sizing of heat transport equipment was conducted to develop CAD of equipment. CFD was preformed on HX1 to validate preliminary sizing modelling. Theoretical results were used to purchase equipment that would maintain *Tj* at 50℃ during nominal load. Equipment purchased was capable of maintain *Tj* at 50℃ at 100% above nominal load and at 30℃ during nominal load. This was a result of the increased fluid velocity and heat transfer area of HX1 compared to theoretical calculations.

This exercise provided the following general insights into the design of a HTS for these applications.

1. The HTS should be designed at the casing ambient temperature
2. The design of the water block, coolant velocity, and coolant temperature are the most significant parameters for the performance of the HTS system.
3. The thermal performance of the radiator is not as significant as the thermal performance of the cooling block. CFD verified that the outlet temperature of the HX1 is virtually the same as the inlet. Therefore, there is minimum cooling required for the fluid when it exists HX1.

Future work includes the design of the radiator considering the finned surface and developing mechanical designs for pumps and fans that are cost competitive with products available on the market.

References

[1] G. P. Towler and R. K. Sinnott, Chemical Engineering Design: Principles, Practice, and Economics of Plant and Process Design, 2013.

[2] K. Thulukkanam, Heat Exchanger Design Handbook, Boca Raton: CRC Press, 2013.

[3] S. T.K., "Estimate friction factor accurately," *Chemical Engineering Journal,* vol. 91, no. 5, pp. 63-64, 1984.

[4] J.F. Louvar & D.A. Crowl., Chemical Process Safety., Pearson Education, 2011.

[5] Intel, "Intel® Core™ i5-8400 Processor," 2020. [Online]. Available: https://ark.intel.com/content/www/us/en/ark/products/126687/intel-core-i5-8400-processor-9m-cache-up-to-4-00-ghz.html. [Accessed 24 September 2020].

[6] T. L. Bergman, A. S. Lavine, F. P. Incropera and D. P. Dewitt, Introduction to Heat Transfer (sixth edition), John Wiley & Sons, 2011.

Appendix 1 – Script

```
# Load packages
library(tidyverse)

# Functions
# function to size the cooling block
size_block <- function(t_block_feed,
            u,
            block_dimensions,
            fin_width,
            fin_height,
            chip_conductivity,
            chip_dimensions,
            t_junction,
            chip_power,
            block_conductivity,
            casing_height,
            rho,
            mu,
            cp,
            k,
            d_i
) {
  n_fins = 1
  q_new <- 0
  while (q_new < chip_power) {
    block_hta <- get_block_hta(block_dimensions,
                n_fins,
                fin_width,
                fin_height)

    s_temp <- get_surface_temp(chip_conductivity,
                chip_dimensions,
                t_junction,
                chip_power,
                block_conductivity,
                block_hta,
                block_dimensions)
    d_hydraulic <- get_hydraulic_diam(casing_height,
                          block_dimensions)
    m <- u*((pi/4)*(d_i)^2)*rho
    u_block <- m/(rho*casing_height*block_dimensions[2])

    h <- get_htc(rho,
         mu,
         d_hydraulic,
         u_block,
         cp,
         k)

    q_new <- -1*h*block_hta*(t_block_feed - s_temp)
    n_fins <- n_fins + 1
  }
  t_block_outlet <- chip_power/(m*cp) + t_block_feed
  return(c(t_block_outlet, n_fins, q_new, m))
}

# function that gets the hta based on the defined block geometry
get_block_hta <- function(block_dimensions,
              n_fins,
              fin_width,
              fin_height) {
  if (block_dimensions[1]*block_dimensions[2] - 
n_fins*fin_width*block_dimensions[2] < 0) {
    stop("The number of fins exceed the area avaliable allocation area, try 
reducing the number of fins.", call. = FALSE)
  } else {
    block_hta <- t_block_outlet <- chip_power/(m*cp) + t_block_feed
  }
  return(block_hta)
}

# function that retrives the surface temperatures of the chip and of the 
block
get_surface_temp <- function(chip_conductivity,
              chip_dimensions,
```

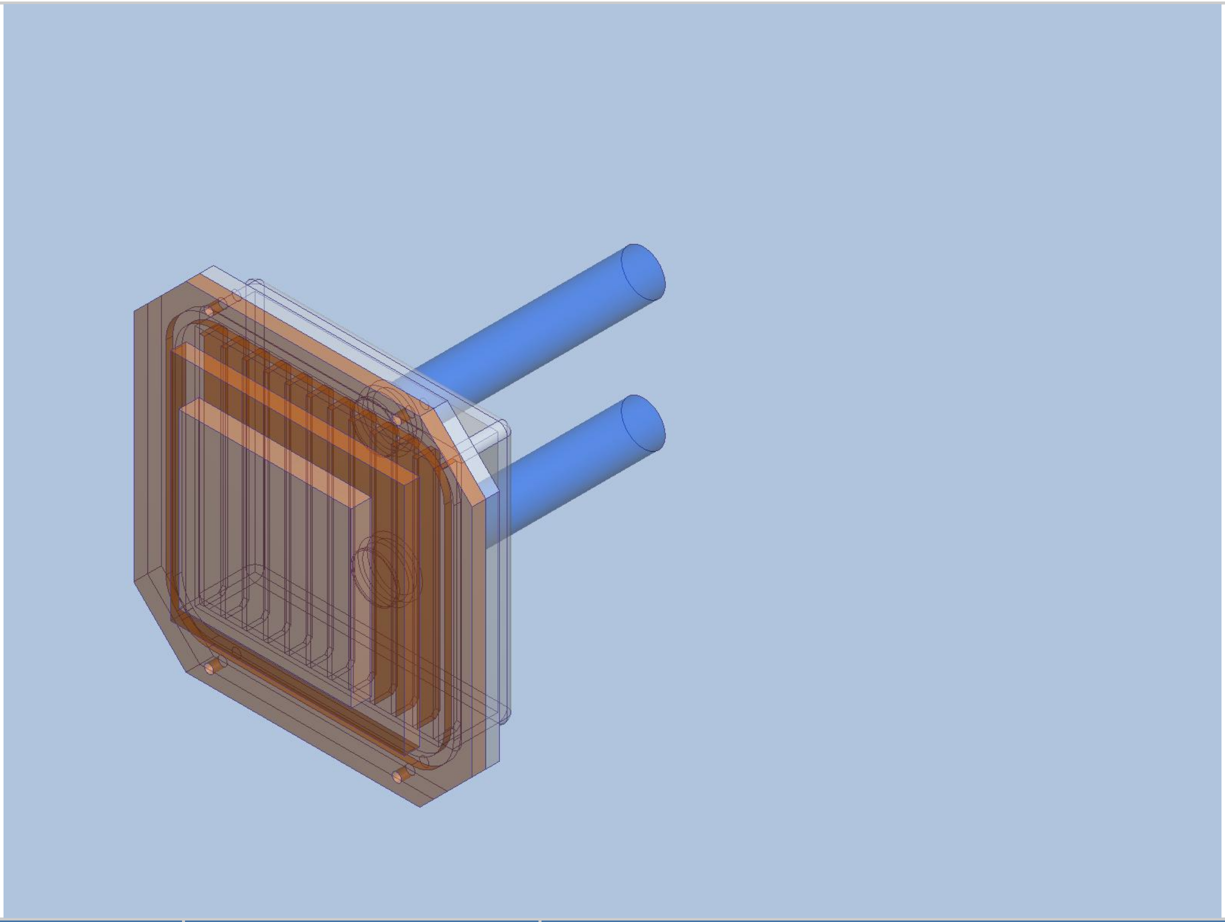
```
                t_junction,

                chip_power,

                block_conductivity,

                block_hta,

                block_dimensions) {

  t_surface1 <- t_junction -
(chip_power*chip_dimensions[3])/((chip_dimensions[1]*chip_dimensions
[2])*chip_conductivity)

  t_surface2 <- t_surface1 -
(chip_power*block_dimensions[3])/((block_hta*block_conductivity))

  return(t_surface2)

}


# function that retrieves the hydralic diameter of the block

get_hydraulic_diam <- function(casing_height,

                               block_dimensions) {

  d_hydraulic <- 2*(casing_height*block_dimensions[2])/(casing_height +
block_dimensions[2])

  return(d_hydraulic)

}


# function that retrieves heat transfer coefficient

get_htc <- function(rho,

                    mu,

                    d_hydraulic,

                    u,

                    cp,

                    k) {

  re <- (rho*u*d_hydraulic)/(mu)

  pr <- (cp*mu)/(k)

  nu <- 0.023*re^0.8*pr^0.33

  h <- k*nu/d_hydraulic

  return(h)

}


# function to size the radiator

size_radiator <- function(t_water_in,

                          t_water_out,

                          m_water,

                          t_air_in,

                          rho_water,

                          mu_water,

                          cp_water,

                          k_water,

                          w_rad,

                          l_rad,

                          l_man,

                          h_rad,

                          r_fan_blade,

                          w_f,

                          mu_air,

                          rho_air,

                          k_air,

                          cp_air,

                          r_motor,

                          q_req,

                          k_wall) {

u_guess <- 760

n_channels <- 1

running <- TRUE

repeat {

  n_channels <- n_channels + 1

  if (n_channels %% 2 == 0) {

    hta_params <- get_hta(n_channels,

                          w_rad,

                          l_rad,

                          h_rad,

                          l_man,

                          m_water,

                          rho_water,

                          w_f)

    n_fins <- hta_params[1]

    hta <- hta_params[2]

    d_hydraulic <- hta_params[3]

    u_water <- hta_params[4]

    l_c <- hta_params[5]

    w_c <- hta_params[6]
```

```
    t_air_out <- 0.0050*u_guess*((t_water_in + t_water_out)/2 - t_air_in)
+ t_air_in

    a_guess <- q_req/(u_guess*(t_air_out - t_air_in))

    m_air <- q_req/(cp_air*(t_air_out - t_air_in))

    air_flowrate <- m_air/rho_air

    u_air <- air_flowrate/(pi*((l_rad/2)^2 - r_motor^2))

    d_o = sqrt((l_c*w_c*8)/((l_c+w_c)*pi))

    d_i = d_o - 0.0001

    htc_water <- get_htc(rho_water,
                         mu_water,
                         d_i,
                         u_water,
                         cp_water,
                         k_water)

    htc_air <- get_htc_air(k_air,
                           d_hydraulic,
                           mu_air,
                           u_air,
                           rho_air,
                           cp_air)

    u_overall <- 1/((1/htc_water) + (1/htc_air) + d_o*log((d_o/d_i),base =
exp(1))/(2*k_wall))

   }

   # IF 0<(U_OVERALL-U_ass)/U_ass<0.3, U_ass <- U_OVERALL

   if (hta < a_guess) {

    u_guess <- u_overall

   } else {

    break

   }

  }

  dp <- get_pressure_drop(l_c,
                          d_i,
                          rho_water,
                          u_water,
                          mu_water)

  return(c(hta, n_fins, n_channels, w_c, u_overall, dp, t_air_out))

}


# function that gets the hta based on the defined radiator geometry

get_hta <- function(n_channels,
                    w_rad,
                    l_rad,
                    h_rad,
                    l_man,
                    m_water,
                    rho_water,
                    w_f) {

  w_c <- w_rad/(2*n_channels - 1)

  l_c <- l_rad - 2*l_man

  afa <- 2*w_c*l_c

  af <- 2*w_c*w_f

  n_fin <- afa/(2*af)*n_channels

  hta <- 2*l_c*w_c*n_channels + 2*l_c*h_rad*n_channels

  d_hydraulic <- (4*(l_rad*w_rad - n_channels*l_c*w_c))/((n_channels -
1)*(2*w_c+l_c))

  u_water <- (m_water)/(rho_water*w_c*h_rad)

  return(c(n_fin, hta, d_hydraulic, u_water, l_c, w_c))

}


# function that gets the heat transfer coefficient for the air side

get_htc_air <- function(k_air,
                        d_hydraulic,
                        mu_air,
                        u_air,
                        rho_air,
                        cp_air) {

  re <- (rho_air*u_air*d_hydraulic)/(mu_air)

  pr <- (cp_air*mu_air)/(k_air)

  nu_for <- 0.158*re^0.66*pr^0.37

  h_air <- (nu_for*k_air)/d_hydraulic

  return(h_air)

}


# function that gets the pressure drop in the radiator

get_pressure_drop <- function(l_c,
                              d_i,
                              rho_water,
```

```
                         u_water,
                         mu_water) {
  re <- (rho_water*u_water*d_i)/(mu_water)
  A <- -2*log10((0.0015/1000)/(3.7*d_i) + 12/re)
  B <- -2*log10((0.0015/1000)/(3.7*d_i) + 2.51*A/re)
  C <- -2*log10((0.0015/1000)/(3.7*d_i) + 2.51*B/re)
  f <- (A -((B-A)^2/(C-(2*B) + A)))^-2
  dp <- 2*(8*f*(l_c/d_i) + 2.5)*(rho_water*u_water^2)/2
  return(dp)
}

# function to get the pump power
get_pump_power <- function(rad_dp,
                         d_i,
                         rho_water,
                         mu_water,
                         u_water,
                         m_water) {
  re <- (rho_water*u_water*d_i)/(mu_water)
  units <- 2*rad_dp*(u_water^2/(2*9.81)) # Assume that dp rad = dp water
block
  entrances_exits <- 3*((160/re + 0.50) + (1.0))*(u_water^2/(2*9.81))
  A <- -2*log10((0/1000)/(3.7*d_i) + 12/re)
  B <- -2*log10((0/1000)/(3.7*d_i) + 2.51*A/re)
  C <- -2*log10((0/1000)/(3.7*d_i) + 2.51*B/re)
  tube_length_f <- ((A -((B-A)^2/(C-(2*B) + A)))^-2)
  tube_length <- ((8*tube_length_f*(2/d_i) +
2.5)*(rho_water*u_water^2)/2)/rho_water # Assume 2m tubing
  pow <- -1*(tube_length + entrances_exits + units)*m_water
}

# Solution
block_params <- size_block(t_block_feed = (273 + 25),
                         u = 0.45,
                         block_dimensions = c(2, 2, 0.25)*(0.0254),
                         fin_width = 0.001,
                         fin_height = 0.005,
                         chip_conductivity = 400,
                         chip_dimensions = c(37.5, 37.5, 4.4)*(1/1000),
                         t_junction = (273 + 50),
                         chip_power = 65,
                         block_conductivity = 400,
                         casing_height = 0.013,
                         rho = 998,
                         mu = 855*10^-6,
                         cp = 4.179*10^3,
                         k = 613*10^-3,
                         d_i = 0.0095)

rad_params <- size_radiator(t_water_in = block_params[1],
                         t_water_out = 298,
                         m_water = block_params[4],
                         t_air_in = 298,
                         rho_water = 998,
                         mu_water = 855*10^-6,
                         cp_water = 4.179*10^3,
                         k_water = 613*10^-3,
                         w_rad = 0.165,
                         l_rad = 0.165,
                         h_rad = 0.020,
                         l_man = 0.020,
                         mu_air = 71.1*10^-7,
                         rho_air = 1.16,
                         k_air = 26.3*10^-3,
                         cp_air = 1.007*10^3,
                         r_motor = 0.05,
                         w_f = 0.001,
                         q_req = block_params[3],
                         k_wall = 613*10^-3)

pump_params <- get_pump_power(rad_dp = rad_params[6],
                         d_i = 0.0095,
                         rho_water = 998,
                         mu_water = 855*10^-6,
                         u_water = 0.45,
                         m_water = block_params[4])
```

Scenario 1

Materials



NAME	ASSIGNED TO	PROPERTIES	
Copper	processor chip:1 water block:1	X-Direction	Piecewise Linear
		Y-Direction	Same as X-dir.
		Z-Direction	Same as X-dir.
		Density	8939.58 kg/m3
		Specific heat	380.718 J/kg-K
		Emissivity	0.6
		Transmissivity	0.0
		Electrical resistivity	1.7e-08 ohm-m
		Wall roughness	0.0 meter

PVC	fittings:2 block casing2:1 fittings:1	X-Direction Y-Direction Z-Direction Density Specific heat Emissivity Transmissivity Electrical resistivity Wall roughness	0.25 W/m-K Same as X-dir. Same as X-dir. 1400.0 kg/m3 1250.0 J/kg-K 0.92 0.0 0.0 ohm-m 0.0 meter
Water	water for cfd:1 water for cfd:2 Volume	Density Viscosity Conductivity Specific heat Compressibility Emissivity Wall roughness Phase	Piecewise Linear 0.001003 Pa-s 0.6 W/m-K 4182.0 J/kg-K 2185650000.0 Pa 1.0 0.0 meter Linked Vapor Material
Silicon Rubber	Volume	X-Direction Y-Direction Z-Direction Density Specific heat Emissivity Transmissivity Electrical resistivity Wall roughness	0.7 W/m-K Same as X-dir. Same as X-dir. 1.7 g/cm3 0.7 J/g-K 0.9 0.0 0.0 ohm-cm 0.0 meter

boundary conditions

TYPE	ASSIGNED TO
Total Heat Flux(65 W)	Surface:5
Velocity Normal(0.45 m/s)	Surface:208
Temperature(298 Kelvin)	Surface:208

Initial Conditions

TYPE	ASSIGNED TO

mesh

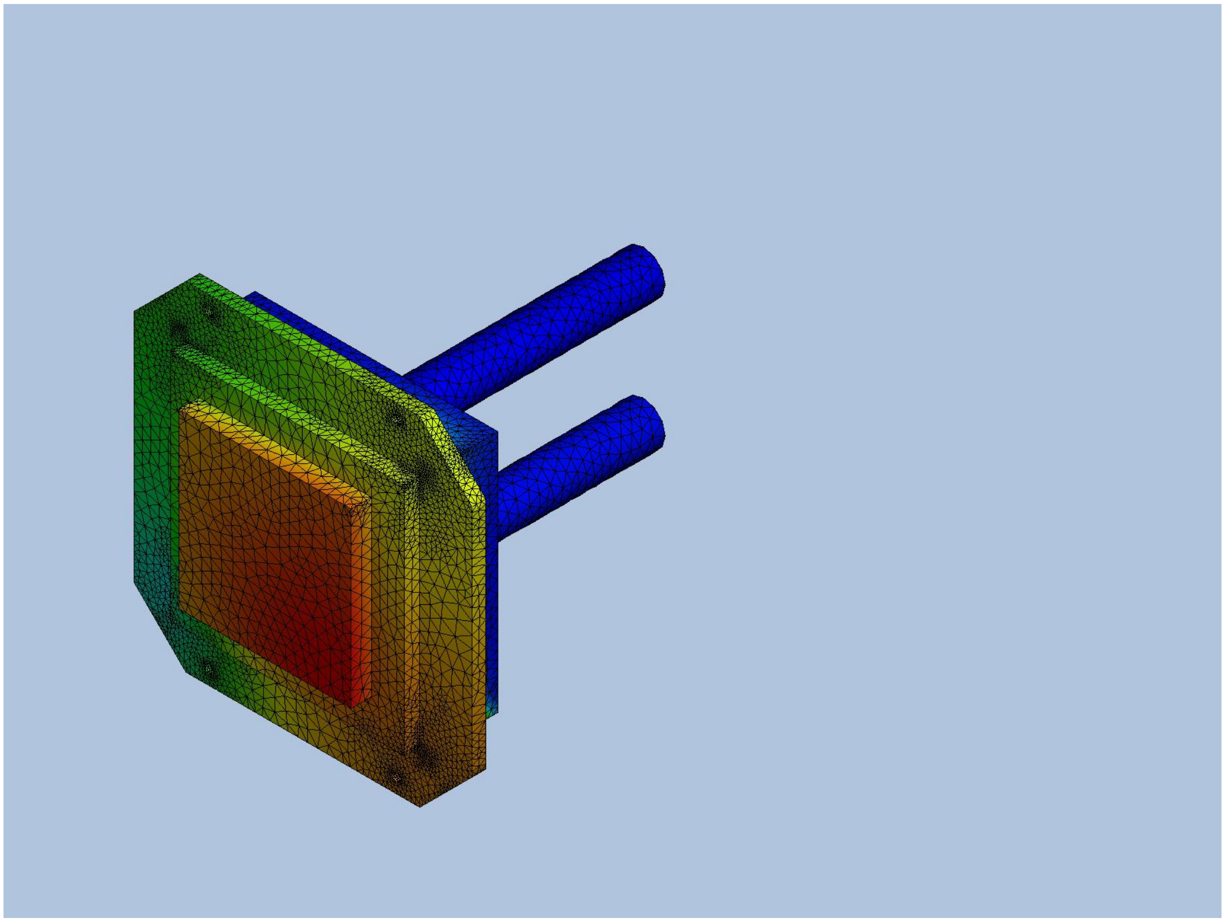
Automatic Meshing Settings

Surface refinement	0
Gap refinement	0
Resolution factor	1.0
Edge growth rate	1.1
Minimum points on edge	2
Points on longest edge	10
Surface limiting aspect ratio	20

Mesh Enhancement Settings

Mesh enhancement	1
Enhancement blending	0
Number of layers	3
Layer factor	0.45
Layer gradation	1.05

Meshed Model



Number of Nodes	98963
Number of Elements	401717

Physics

Flow	On
Compressibility	Incompressible
Heat Transfer	On
Auto Forced Convection	Off
Gravity Components	0.0, 0.0, 0.0
Radiation	Off
Scalar	No scalar
Turbulence	On

Solver Settings

Solution mode	Steady State
Solver computer	MyComputer
Intelligent solution control	On
Advection scheme	ADV 5
Turbulence model	k-epsilon

Convergence

Iterations run	100
Solve time	839 seconds
Solver version	19.2.20190802

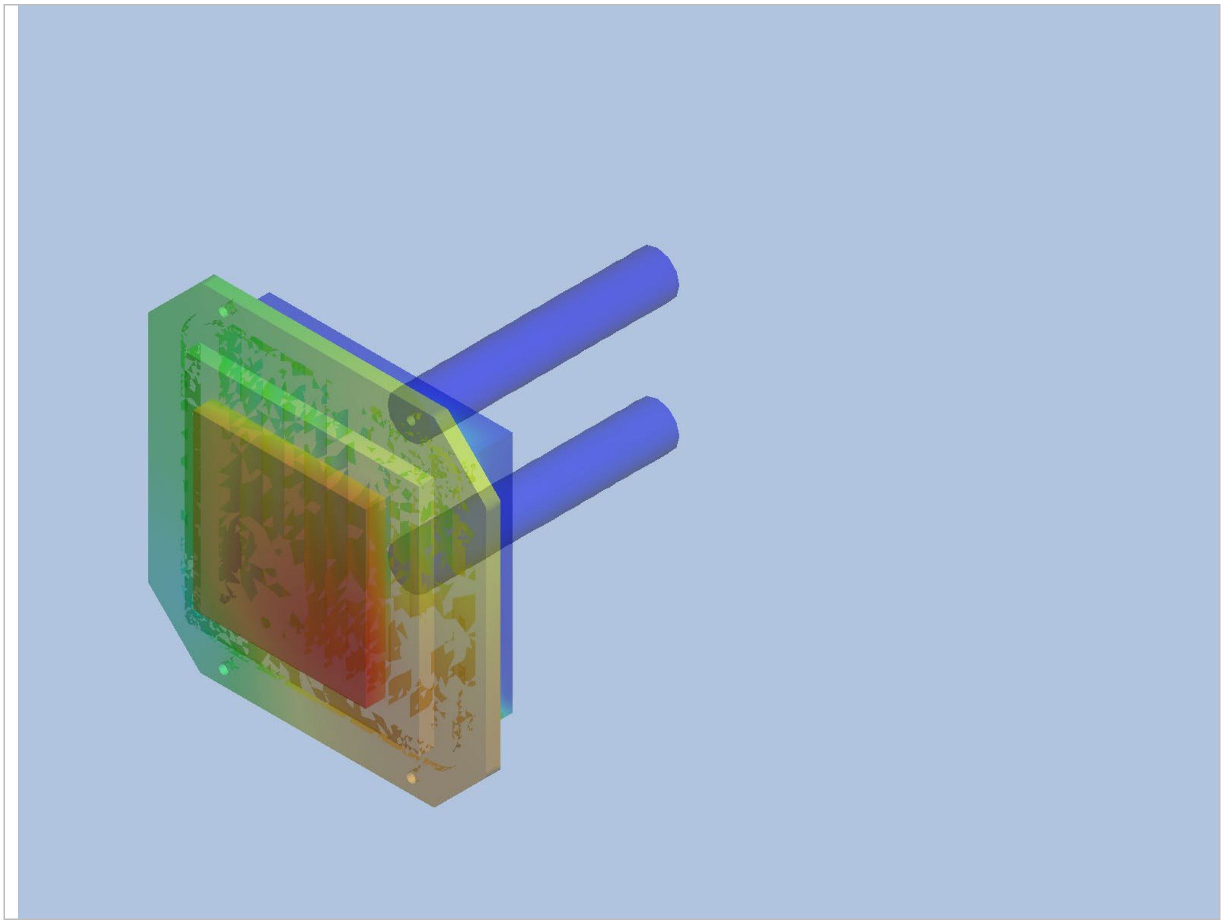
Energy Balance

Fluid Energy Balance Information	(numerical) energy out - heat transfer due to sources heat transfer from wall to mdotin x cp x (tout - tin)	-0.004432 Watts 0.0 Watts 64.881 Watts -37655.0 Watts
Solid Energy Balance Information	heat transfer due to sources heat transfer from exterior heat transfer from fluid to	0.0 Watts 65.0 Watts -64.831 Watts

Mass Balance

	IN	OUT
Mass flow	30.2146 g/s	N.A.
Volume flow	30.2691 cm^3/s	N.A.

Results



Inlets and Outlets

inlet 1	inlet bulk pressure	4.60472e+12
	inlet bulk	24.85 C
	inlet mach number	9.46112e-09
	mass flow in	30.2146 g/s
	minimum x,y,z of	0.0
	node near minimum	6974.0
	reynolds number	3661.52
	surface id	208.0
	total mass flow in	30.2146 g/s
	total vol. flow in	30.2691 cm^3/s
	volume flow in	30.2691 cm^3/s

Field Variable Results

VARIABLE	MAX	MIN
cond	3.85696 W/cm-K	0.006 W/cm-K

dens	8.93958 g/cm^3	0.9982 g/cm^3
econd	1352.88 W/cm-K	0.0 W/cm-K
emiss	1.0	0.0
evisc	336.565 g/cm-s	0.0 g/cm-s
gent	70932900.0 1/s	5.26704e-05 1/s
press	4.60704e+12 dyne/cm^2	4.44862e+12 dyne/cm^2
ptotl	4.61302e+12 dyne/cm^2	0.0 dyne/cm^2
scal1	0.0	0.0
seebeck	0.0 V/K	0.0 V/K
shgc	0.0	0.0
spech	4.182 J/g-K	0.380718 J/g-K
temp	29.4762 C	24.85 C
transmiss	0.0	0.0
turbd	8.70006e+15 cm^2/s^3	0.000319907 cm^2/s^3
turbk	3477530000.0 cm^2/s^2	9.24132e-06 cm^2/s^2
ufactor	0.0	0.0
visc	0.01003 g/cm-s	0.0 g/cm-s
vx vel	31189.4 cm/s	-107245.0 cm/s
vy vel	54346.2 cm/s	-86036.6 cm/s
vz vel	21048.7 cm/s	-112434.0 cm/s
wrough	0.0 cm	0.0 cm

Component Thermal Summary

PART	MINIMUM TEMPERATURE	MAXIMUM TEMPERATURE	VOLUME AVERAGED TEMPERATURE
processor chip:1	27.3843	29.4762	28.6546
fittings:2	0	0	0
block casing2:1	0	0	0
fittings:1	0	0	0
water block:1	24.8533	29.0118	27.7968
water for cfd:1	24.85	24.8503	24.85
water for cfd:2	24.9167	24.9326	24.9298
Volume	24.85	28.838	25.7027
Volume	0	0	0

Fluid Forces on Walls

pressx	70433000.0 dynes
pressy	-3.1168e+12 dynes
pressz	87068000.0 dynes
shearx	-15513.0 dynes
sheary	-10247.0 dynes
shearz	-14851.0 dynes