Arianne 5 Rocket: What Went Wrong?

source: http://en.wikipedia.org/wiki/Ariane_5_Flight_501

The Ariane 5 reused the inertial reference platform from the Ariane 4, but the Ariane 5's flight path differed considerably from the previous models. Specifically, the Ariane 5's greater horizontal acceleration caused the computers in both the back-up and primary platforms to crash and emit diagnostic data misinterpreted by the autopilot as spurious position and velocity data. Pre-flight tests had never been performed on the inertial platform under simulated Ariane 5 flight conditions so the error was not discovered before launch. During the investigation, a simulated Ariane 5 flight was conducted on another inertial platform. It failed in exactly the same way as the actual flight units.

The greater horizontal acceleration caused a data conversion from a 64-bit floating point number to a 16-bit signed integer value to overflow and cause a hardware exception. Efficiency considerations had omitted range checks for this particular variable, though conversions of other variables in the code were protected. The exception halted the reference platforms, resulting in the destruction of the flight.

Although the report identified a software bug as the direct cause, other investigators see the causes as system design failures and management issues:[4][5]

> h) On the basis of those calculations the main computer commanded the booster nozzles, and somewhat later the main engine nozzle also, to make a large correction for an attitude deviation that had not occurred.
> i) A rapid change of attitude occurred, which caused the launcher to disintegrate at 39 seconds after H0 due to aerodynamic forces.
> m) Ariane 5's inertial reference system is essentially the same as a system presently flying on Ariane 4. The part of the software that caused the interruption in the inertial system computers is used before launch to align the inertial reference system and, in Ariane 4, also to enable a rapid realignment of the system in case of a late hold in the countdown. This realignment function, which does not serve any purpose on Ariane 5, was nevertheless retained for commonality reasons and allowed, as in Ariane 4, to operate for approx. 40 seconds after lift-off.
> n) During design of the software of the inertial reference system used for Ariane 4 and Ariane 5, a decision was taken that it was not necessary to protect the inertial system computer from being made inoperative by an excessive value of the variable related to the horizontal velocity, a protection provided for several other variables of the alignment software. When taking this design decision, it was not analysed or fully understood which values this particular variable might assume when the alignment software was allowed to operate after lift-off.
> o) In Ariane 4 flights using the same type of inertial reference system there has been no such failure because the trajectory during the first 40 seconds of flight is such that the particular variable related to horizontal velocity cannot reach, with an adequate operational margin, a value beyond the limit present in the software.
> p) Ariane 5 has a high initial acceleration and a trajectory, which leads to a build-up of horizontal velocity five times more rapid than for Ariane 4. The higher horizontal

velocity of Ariane 5 generated, within the 40-second timeframe, the excessive value that caused the inertial system computers to cease operation.

q) The purpose of the review process, which involves all major partners in the Ariane 5 programme, is to validate design decisions and to obtain flight qualification. In this process, the limitations of the alignment software were not fully analysed and the possible implications of allowing it to continue to function during flight were not realised.

r) The specification of the inertial reference system and the tests performed at equipment level did not specifically include the Ariane 5 trajectory data. Consequently the realignment function was not tested under simulated Ariane 5 flight conditions, and the design error was not discovered.

s) It would have been technically feasible to include almost the entire inertial reference system in the overall system simulations which were performed. For a number of reasons it was decided to use the simulated output of the inertial reference system, not the system itself or its detailed simulation. Had the system been included, the failure could have been detected.

t) Post-flight simulations have been carried out on a computer with software of the inertial reference system and with a simulated environment, including the actual trajectory data from the Ariane 501 flight. These simulations have faithfully reproduced the chain of events leading to the failure of the inertial reference systems

Arithmetic overflow

According to a presentation by Jean-Jacques Levy (who was part of the team who searched for the source of the problem), the source code in Ada that caused the problem was as follows.[6]

Original Code

```
L_M_BV_32 := TBD.T_ENTIER_32S ((1.0/C_M_LSB_BV) *
G_M_INFO_DERIVE(T_ALG.E_BV));

if L_M_BV_32 > 32767 then
    P_M_DERIVE(T_ALG.E_BV) := 16#7FFF#;
elsif L_M_BV_32 < -32768 then
    P_M_DERIVE(T_ALG.E_BV) := 16#8000#;
else
    P_M_DERIVE(T_ALG.E_BV) := UC_16S_EN_16NS(TDB.T_ENTIER_16S(L_M_BV_32));
end if;

P_M_DERIVE(T_ALG.E_BH) :=
  UC_16S_EN_16NS (TDB.T_ENTIER_16S ((1.0/C_M_LSB_BH) *
G_M_INFO_DERIVE(T_ALG.E_BH)));
```

The last line (shown here as two lines of text) caused the overflow, where the conversion from the 64 bit float to 16 bit unsigned integer is not protected. The correct code would have been:

```
L_M_BV_32 := TBD.T_ENTIER_32S ((1.0/C_M_LSB_BV) *
G_M_INFO_DERIVE(T_ALG.E_BV));

if L_M_BV_32 > 32767 then
    P_M_DERIVE(T_ALG.E_BV) := 16#7FFF#;
elsif L_M_BV_32 < -32768 then
    P_M_DERIVE(T_ALG.E_BV) := 16#8000#;
else
    P_M_DERIVE(T_ALG.E_BV) := UC_16S_EN_16NS(TDB.T_ENTIER_16S(L_M_BV_32));
end if;

L_M_BH_32 := TBD.T_ENTIER_32S ((1.0/C_M_LSB_BH) *
G_M_INFO_DERIVE(T_ALG.E_BH));

if L_M_BH_32 > 32767 then
    P_M_DERIVE(T_ALG.E_BH) := 16#7FFF#;
elsif L_M_BH_32 < -32768 then
    P_M_DERIVE(T_ALG.E_BH) := 16#8000#;
else
    P_M_DERIVE(T_ALG.E_BH) := UC_16S_EN_16NS(TDB.T_ENTIER_16S(L_M_BH_32));
end if;
```

in other words, the same overflow check should have been present for the horizontal part of the calculation (E_BH) as was already present for the vertical part of the calculation (E_BV).

Aftermath

Following the failure, four replacement Cluster II satellites were built. These were launched in pairs aboard Soyuz-U/Fregat rockets in 2000.

The launch failure brought the high risks associated with complex computing systems to the attention of the general public, politicians, and executives, resulting in increased support for research on ensuring the reliability of safety-critical systems. The subsequent automated analysis of the Ariane code was the first example of large-scale static code analysis by abstract interpretation.[7]

The failure also harmed the excellent success record of the European Space Agency's rocket family, set by the high success rate of the Ariane 4 model. It was not until 2007 that Ariane 5 launches were recognised as being as reliable as those of the predecessor model.[8]